THE WORLD OF DOCUMENTS

EUROSCRIPT DOCUMENTUM REPORT GENERATOR

eDRG 3.6 User Guide Documentum Administrator, Webtop & TaskSpace

Deliver meaningful insight across your organisation

Contents

1 Product Description

With the euroscript Documentum Report Generator, end-users can request reports from predefined report definitions. They can see the reports displayed on their computer screen or receive reports automatically in their e-mail or at a specified location inside or outside the Documentum repository. The preferred format can be chosen by the end-user, without the end-user needing any knowledge about DQL, HTML or XSL.

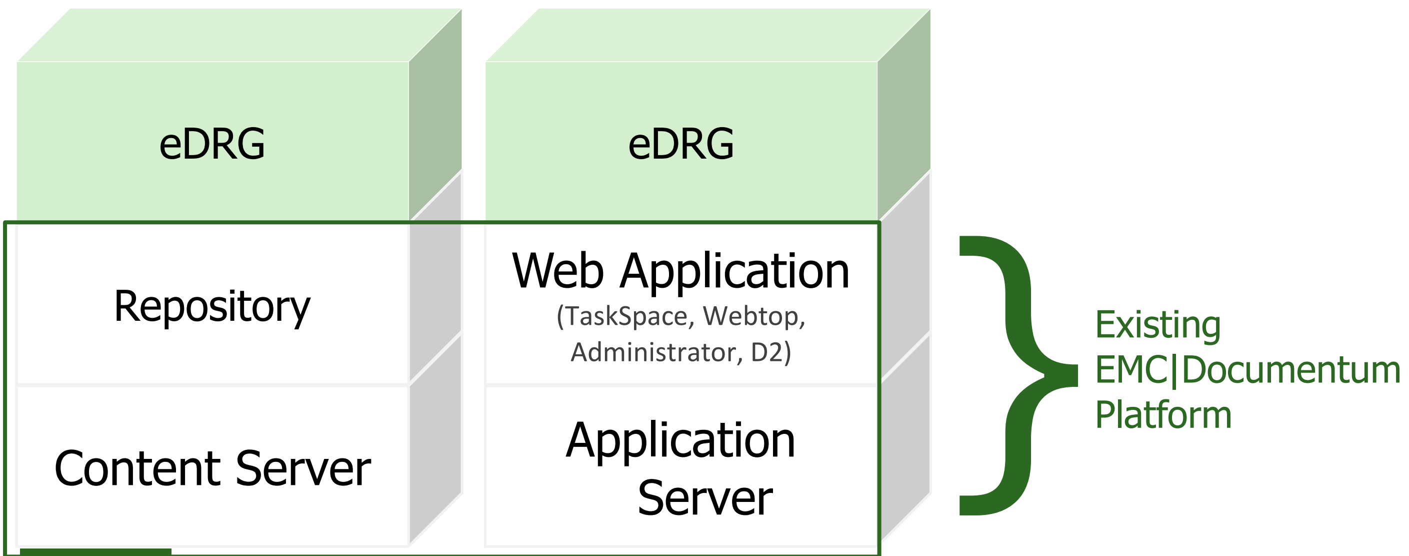
The euroscript Documentum Report Generator categorizes functionalities according to the user's role in the report generator. There are 3 predefined roles:

- Users in the *Report User* role, which supplies predefined reports from the system, require no knowledge of either DQL or style sheets.
- Users in the *Report Administrator* role can schedule reports so that they are generated automatically at a predefined time and location.
- Users in the *Report Builder* role can define new report specifications by configuring the DQL statements, and they can identify and compose the desired style sheets.

The euroscript Documentum Report Generator can be accessed by anyone who has been given one of these default roles.

The euroscript Documentum Report Generator can produce output in any format that can be generated using XSL style sheets. For example: reports can be presented in PDF, HTML, text file, Microsoft Excel spread sheet or Microsoft Word format. The euroscript Documentum Report Generator can also e-mail the output file automatically. In this case the recipient does not need to be a Documentum user and can even be someone from outside the organisation.

The euroscript Documentum Report Generator is integrated into the Documentum Webtop, Documentum TaskSpace and Documentum Administrator client, in line with the corporate philosophy of Documentum. This enables end users to work in an environment that they are already familiar with. Besides integration in these environments, euroscript has also integrated the euroscript Documentum Report Generator in the popular Documentum D2-client and D2 4.0. Due to its full integration within the Documentum environment, no additional components need to be installed on the end user's local machine.



As shown in the figure on the previous page, the euroscript Documentum Report Generator components are located on both Repository level as Web Application Level. Both use standard EMC | Documentum subcomponents for its functionality. In pre euroscript Documentum Report Generator 3.0 versions, also some components where needed on the Java Method Server. As from release 3.1 this is no longer required.

Since 2012 the report generator has been EMC certified. In this same year, euroscript has become an EMC select partner. Because of this the name has been changed from euroscript Delt Report Generator to euroscript Documentum Report Generator. In the product, you will still find some references to the old name. This will be adjusted in a future release.

2 Introduction

2.1 Roles

The euroscript Documentum Report Generator uses three predefined roles to allow access to the functionality it provides. After the euroscript Documentum Report Generator has been installed, users have to be assigned to each of these roles, as appropriate, using Documentum Administrator.
Note that these roles are hierarchical; e.g. a report_builder is also a report_user and a report_administrator is a report_builder and a report_user as well.

- **report_user**
  This role is used by the euroscript Documentum Report Generator for users that are allowed to generate reports and view report history.

- **report_builder**
  This role is used by the euroscript Documentum Report Generator for users that are allowed to create new report definitions and manage report presentations.

- **report_administrator**
  This role is used by the euroscript Documentum Report Generator for users that are report administrators, which are allowed to manage report schedules and manage report structure.

Functionality for each role:

	Report User	Report Builder	Report Administrator
Run reports	✓	✓	✓
Show reports history	✓	✓	✓
Create report from Objects	✓	✓	✓
Create report definitions	✘	✓	✓
Copy/move report definitions	✘	✓	✓
Copy/move report categories	✘	✓	✓

Update report categories	✗	✓	✓
Create report presentations	✗	✓	✓
Copy/move report presentations	✗	✓	✓
Delete report presentations	✗	✓	✓
Delete report definitions	✗	✗	✓
Create report categories	✗	✗	✓
Delete report categories	✗	✗	✓
Create report schedules	✗	✗	✓
Delete report schedules	✗	✗	✓
Check reports history tables	✗	✗	✓
Enable/disable allow historic data	✗	✗	✓[1]
Configure Delt Report Agent job	✗	✗	✓[2]

1 If the user is a member of the admingroup.

2 If the user has access to Documentum Administrator and permissions to configure jobs.

NOTE

These functionalities are available as long as the objects have their default permission set. If the permissions are changed, some functions might no longer be available.

2.2 Reports

Report objects are the reports generated by either a schedule or a manual request. These objects can be stored anywhere in the repository and can be used as any other document.

2.3 Report Definitions

Report Definitions are used to generate reports. You can access the Report Definitions from the Report Generator tree node. A report definition must be placed in a report category and can be created by a report builder.

2.4 Report Categories

Report Categories are used to group report definitions. Each report category can contain other report categories and report definitions. Each report definition must be placed in a report category. Report categories can be created by a report administrator.

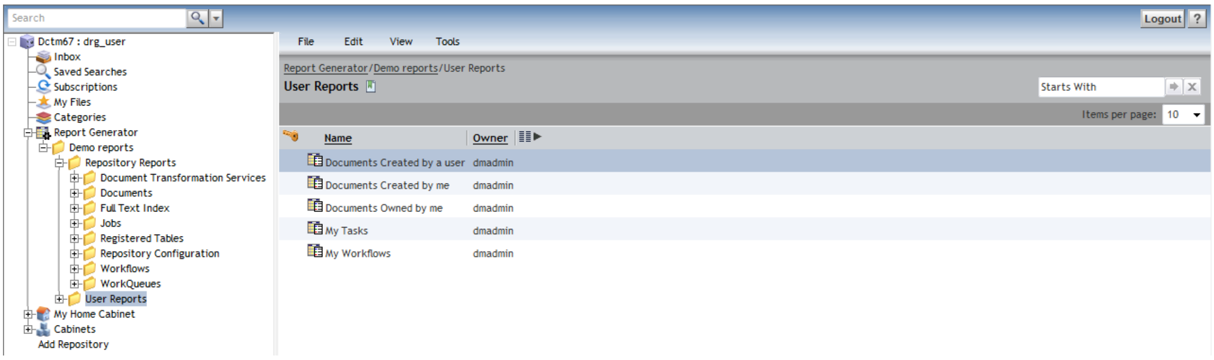
2.5 Report Schedules

Report Schedules are used to schedule reports for automatic generation.
Example: using a schedule, a report can be generated every Saturday at 23:00 or every day at any other desired time.

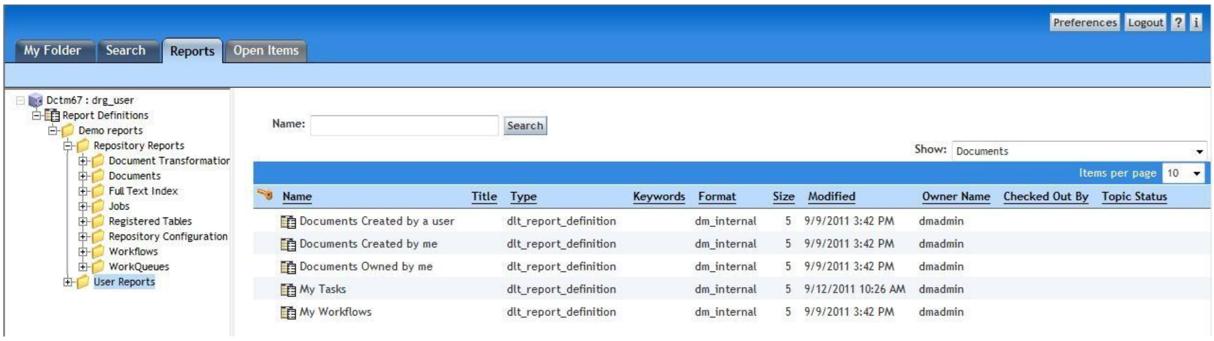
3 Report User

3.1 Introduction

This chapter describes the functionality for users with the report_user role. Users with the report_builder or report_administrator role, also have the report_user role.



↳ Report Generator tree node in Documentum Administrator and Webtop.

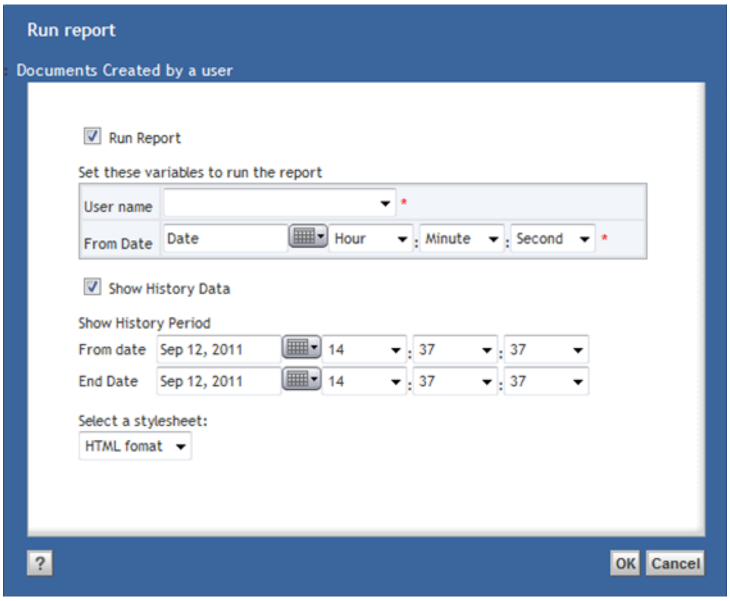


↳ Report Generator tree node in Documentum TaskSpace.

3.2 Generating Reports

Double-click on a Report Definition to generate the report.
After the report is generated, it is placed in your home cabinet and automatically opened in the application for the file type of the report; e.g. Adobe Acrobat Reader for PDF reports.

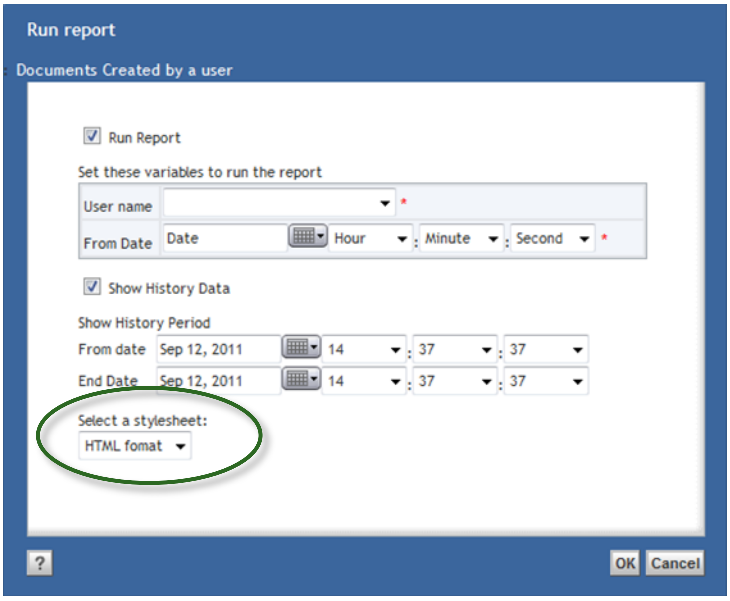
If the Report Definition needs more information to generate the report, a new dialog will be displayed. The following paragraphs describes this dialog.



↳ The Run Report dialog.

3.2.1 Choosing a Stylesheet

If a Report Definition has multiple output definitions (stylesheets), a dropdown box with the available stylesheets will be displayed.

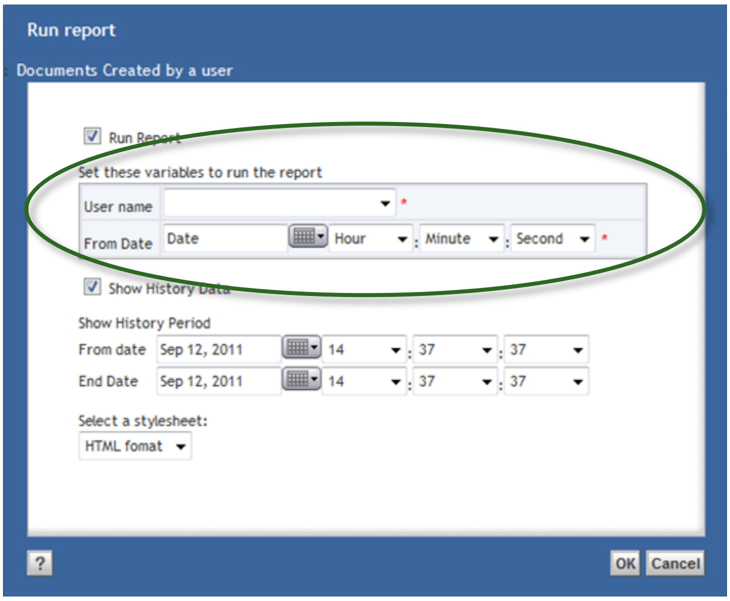


↳ Dropdown box with available stylesheets.

3.2.2 Setting Variables

A Report Definition can have optional or required, and single or repeating variables. You can/must specify values for these variables when generating a report. The following paragraphs describe the types of variables that can be set.

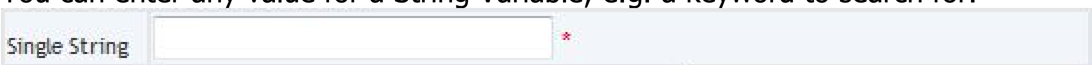
Repeating variables have an [**Add**] and [**Remove**] button to add or remove values from the list.



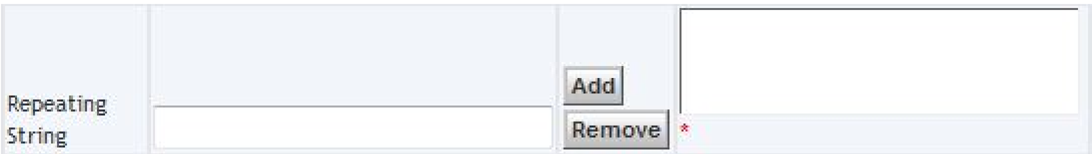
↳ The variables.

3.2.2.1 String Variables

You can enter any value for a String Variable; e.g. a keyword to search for.



↳ Single String Variable.



↳ Repeating String Variable.

3.2.2.2 Date Variables

You can use the date selector and time drop down boxes to set a date for a Date Variable; e.g. oldest date to include.

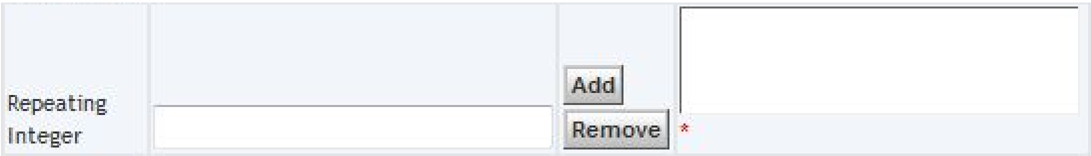
↳ Single Date Variable.

There are no repeating date variables.

3.2.2.3 Integer Variables

You can enter any number for Integer Variables. If you enter a text that is not a number, you will get an error message.

Single Integer Variable.

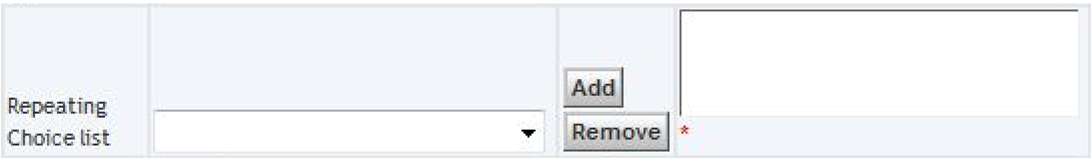


Repeating Integer Variable.

3.2.2.4 Choice List Variables

For Choice List Variables you can select a value from the dropdown box; e.g. a user name.

Single Choice List Variable.

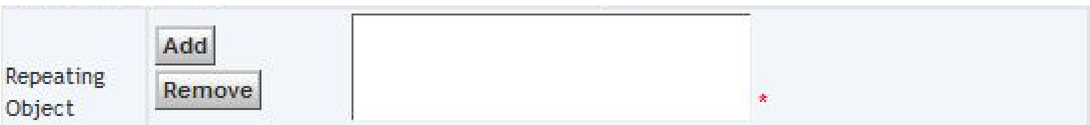


Repeating Choice List Variable.

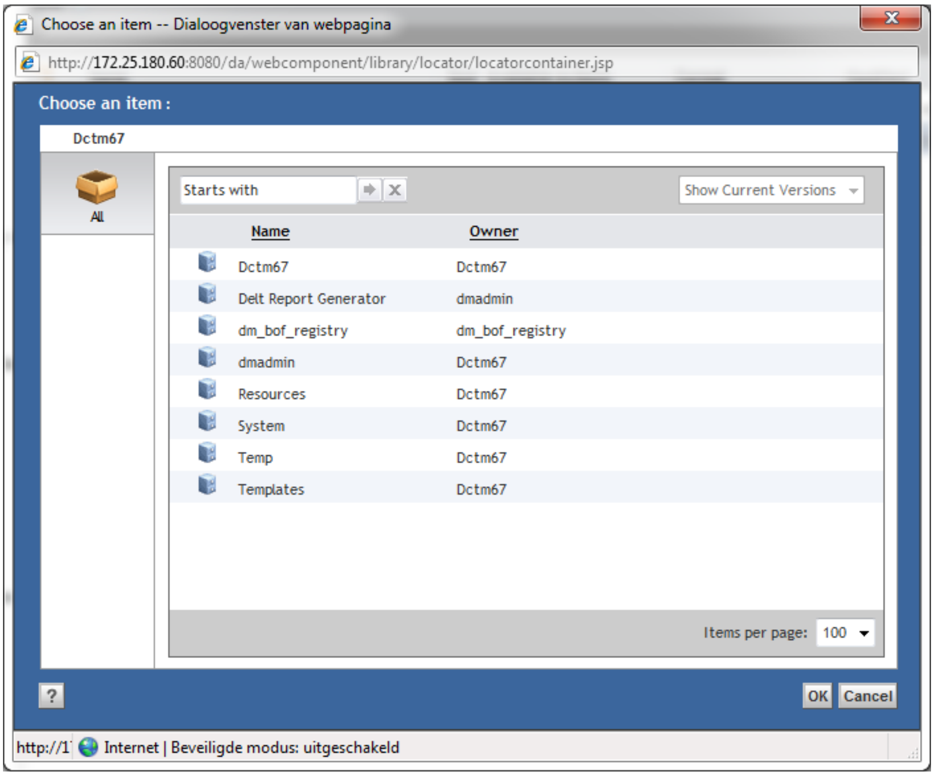
3.2.2.5 Object Variables

Use the [**Add**] button to select an object from the repository for the Object Variable. The allowed object types to select, are set by the Report Builder; e.g. a folder to search in.

Single Object Variable.



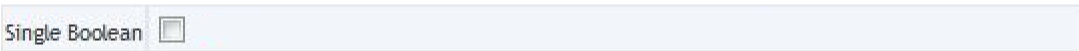
Repeating Object Variable.



↳ The object browser is shown when clicking the [**Add**] button.

3.2.2.6 Boolean Variables

You can use the checkbox to enter true or false for the Boolean Variable; e.g. to include or exclude older versions.

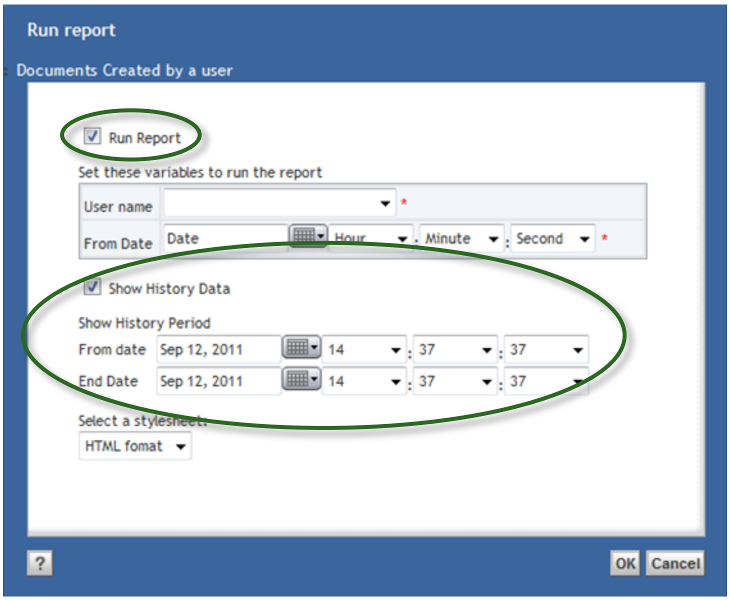


↳ Single Boolean Variable.

There are no repeating Boolean Variables.

3.2.3 Including Historic Data

If a Report Definition is enabled to store and process historic data, you can choose to include it in the report you are generating.
There are 2 options you can set for the historic data.



The first checkbox, Run Report, is used to indicate if the queries that are set on this Report Definition should run or not. If the checkbox is not checked, you will only get historic data. If the checkbox is checked, you will also get the current data.
The second checkbox, Show History Data, is used to indicate if historic data should be included in the report. The two date selectors below this checkbox can be used to specify the period you want to be included in the historic data.

3.3 Subscribing to Report Definitions

You can subscribe to Report Definitions:

- Select the Report Definition.
- Choose menu item **Tools**.
- Choose submenu item **Subscribe**.

If you double-click on a subscription of a report definition, the report will be generated.

3.4 Subscribing to Report Categories

You can subscribe to Report Categories:

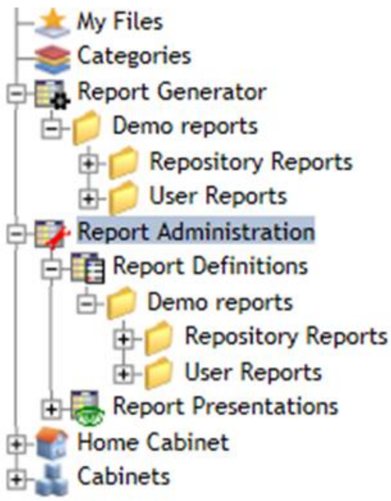
- Select the Report Category.
- Choose menu item **Tools**.
- Choose submenu item **Subscribe**.

4 Report Builder

4.1 Introduction

This chapter describes the functionality for users with the report_builder role. Users with the report_administrator role, also have the report_builder role.
Report Builders can access the "Report Administration" tree node and the underlying tree node "Report Definitions" and "Report Presentations".

The "Report Definitions" tree node contains the same Report Categories and Report Definitions as the "Report Generator" tree node but here the Report Builder can also create new Report Definitions. The "Report Presentations" tree node contains the presentation files for the report definitions.



4.2 Copying/Moving Report Categories

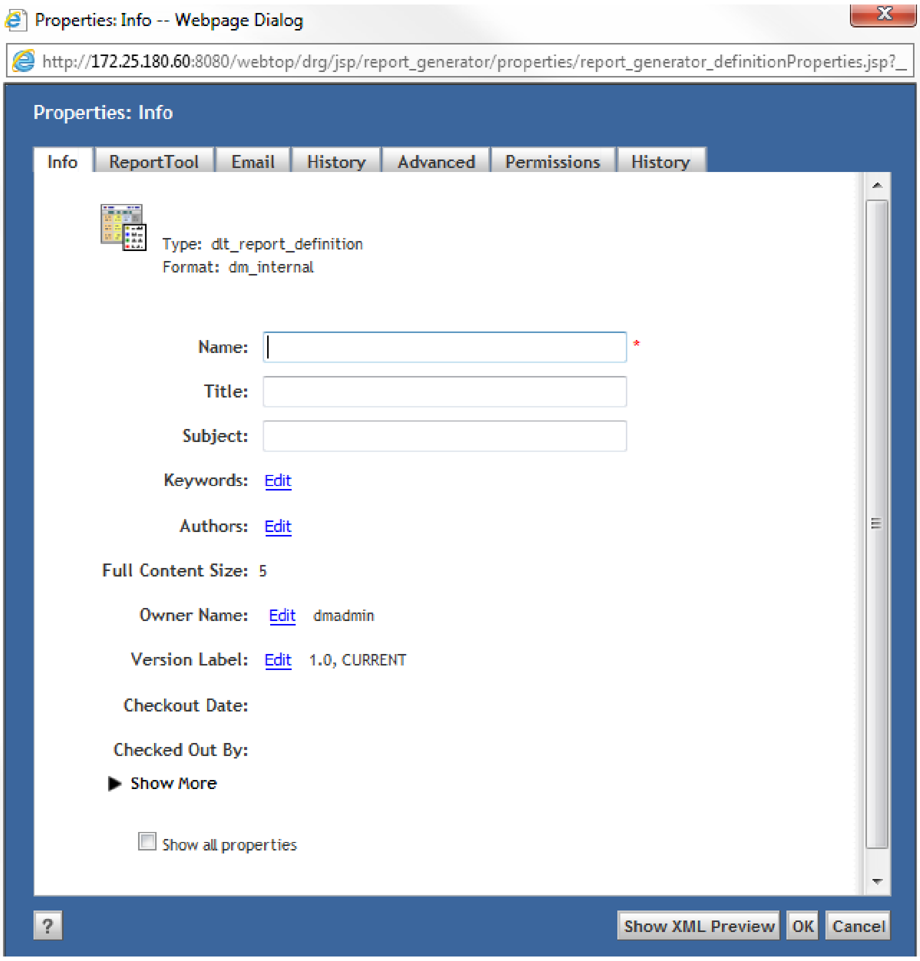
Copying and moving Report Categories works in the same way as copying or moving other objects.
It is only possible to copy or move Report Categories to another report category from **Report Administration\Report Definitions**.

4.3 Creating Report Definitions

Report definitions can be created by Report Builders from the following location:
Report Administration\Report Definitions (not from the Report Generator section).

To create a report definition:

- Navigate to the report category in which you want to create the Report Definition.
- Choose the menu item **File**.
- Choose the submenu item **New**.
- Choose the submenu item **Report Definition**.
  - ↳ The following screen is displayed:

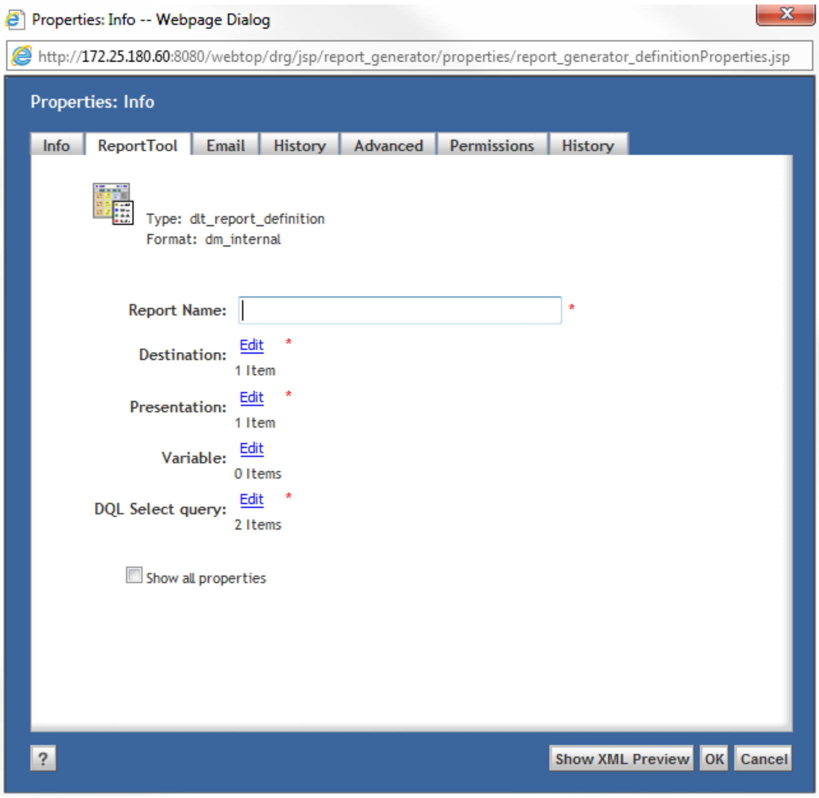


- Fill in a name for the Report Definition.

NOTE

The Report Definition name is subject to the regular object_name attribute restrictions; e.g. it has a maximum length of 255 characters.

- Click [**Next**].



↳ The attributes shown on the info tab in the ReportTool section and following sections, define the reports that will be created by this Report Definition.

- Fill in the required attributes.
- Click [**Finish**].

↳ The Report Definition is created.

Explanation of the attributes:

Attribute	Description	Limits
Date Format 6)	The format that should be used for displaying dates in filenames and results.	32 characters, should be a valid java date pattern 4)
Destination *)	The destinations for the reports. See paragraph 4.3.1.	1000 characters
Destination Type 5) *)		
DQL Select query *)	The DQL queries to run. See paragraph 4.3.2 and paragraph 0.	2000 characters
DQL Name 5) *)		255 characters
Result Threshold 5) *)		
Save Report 5)		
DQL History Table 8)		
Split repeating values	If selected, the repeating values returned from the DQL queries will be split. See paragraph 4.9.1.	
Email Body 1)	The text that is used in emails sent by the Report Generator. This text can contain HTML.	2000 characters
Email Sender 2)	The from-address used in emails	255 characters

	sent by the Report Generator.	
Email Subject 3)	The subject used in emails sent by the Report Generator.	255 characters
Float Format 7)	The format that should be used for displaying numbers in the results.	32 characters, should be a valid java float pattern 4)
Last Status	The last status for this report definition reported by the Reporting Agent.	Read-only field
Presentation *)	The presentations for the reports. See paragraph 4.3.10.	1000 characters
Presentation Name 5) *)		32 characters
Presentation Type 5) *)		
Presentation Format 5) *)		
Presentation Destination 5) *)		32 characters (e.g. max 11-13 destinations)
Variable *)	The variables for the reports. See paragraph 4.3.4.	32 characters
Variable Label 5)		255 characters
Variable Type 5) *)		
Variable Default Value 5)		255 characters
Variable Query 5)		1000 characters
Variable Repeating 5)		
Variable Mandatory		
Report Name *)	The name used for the generated reports. If the report is placed in an OS location or if 'Save as Version' is false, the current date and time are added to the name.	255 characters
Save as Version	If selected, new reports will be saved as new versions of a previous report.	
Save as Rendition	If selected, multiple presentations will be saved as renditions of each other.	
Allow manual run after	Users are only allowed to run this report definition between the specified times. If the 2 values are the same, a manual run is always allowed. These settings do not affect schedules.	
Allow manual run before		
Remove Reports after Number of Versions	Removes reports after the specified number of versions. See paragraph 4.4.	
Remove Reports after Number of Days	Removes old reports after the specified number of days. See paragraph 4.4.	
Use disk cache	If selected the Report Generator will use a disk cache (less memory	

	consumption) instead of a memory cache (faster processing) for temporary files. This option is recommended for large reports.	
Save Run Mode *)	These attributes are used for reports with historic data. See paragraph 0.	
Save Report Interval *)		
Save Report Mode *)		
Retention Interval *)		
Retention Mode *)		
Default Show History *)		
Default Show History Interval *)		
Default Show History Period *)		
"User can select destination"	Indicates if user can choose a destination (location in the repository or email address) when running a report.	

1) If left empty then the following text will be used "This is a report automatically generated by the Delt Report Generator"

2) If left empty then the following text is used "<user_name>@<repository_name>.com"

3) If left empty then the following text will be used "Report from repository"

4) See Appendix I. Java formats

5) These attributes are hidden and won't be displayed on the attributes page but can be set by clicking on the edit link for the visible attribute in the same group (i.e. for destinations the attribute destination is visible and the attribute destination_type is not)

6) If left empty then the following format is used "yyyy/MM/dd hh:mm:ss"

7) If left empty then the following format is used "#,###.##"

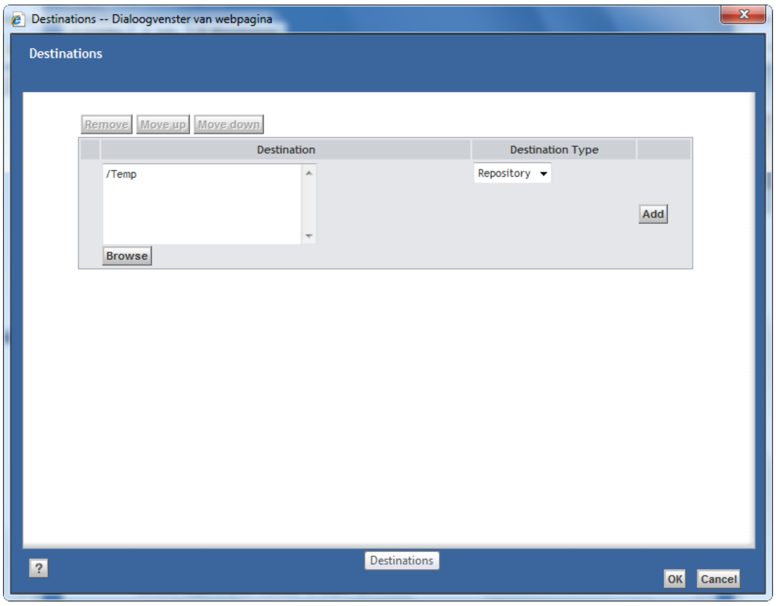
8) This attribute is hidden on both the properties screen and the dql edit screen but is automatically set internally.

*) These attributes are required

4.3.1 Destinations

Destinations define the location where the report will be saved to (or emailed to if the destination type is E-mail). A Report Definition can have multiple destinations.
Each destination specification consists of a Destination and a Destination Type. Only the attribute Destination is displayed on the attributes screen. These two attributes are edited together.

- Click [**Edit**].
  ↳ The edit screen opens:



↳ The edit destinations.

To add new entries:

- Fill in the fields in the bottom row.
- Click [**Add**].

Destination Type	Destination
OS	Must be an existing, accessible, location on the file system on the server with the Java Method Server.
Repository	Must be an existing folder path in the repository.
Email	Must be a valid email address. Multiple email addresses can be specified, separated by a ;.

The Browse button lets the user select a cabinet or folder in the repository as destination. This button only works if Destination Type is set to Repository. If the Destination Type is set to any other value, the button will be disabled.

To edit an existing entry:

- Click [**Edit**] for that row.
- Edit the existing entry.
- Click [**OK**].

If you select more rows:

- Move the rows up or down with the [**Move up**] and [**Move down**] buttons.

Or

- Remove the rows with the [**Remove**] button.
- Click [**OK**].
  ↳ All changes are saved.

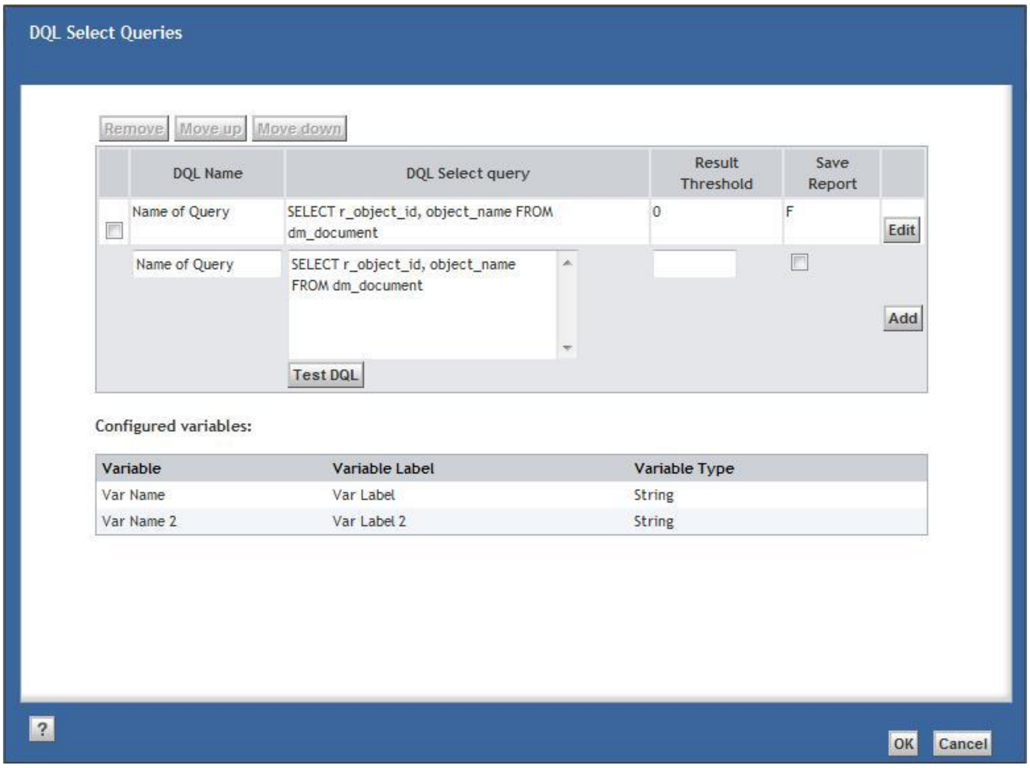
Always specify at least one destination in order to be able to specify presentations.

Destinations are used for automatically generated reports. Manually generated reports are always placed in the user's home cabinet, regardless of any destinations specified.

4.3.2 DQL Select Queries

DQL Select queries define the DQL queries that should be run to build the report. Each report definition can have multiple DQL queries.
A DQL query consists of a DQL Select query, a DQL Name, a Result Threshold and a Save Report. Only the attribute DQL Select query is displayed on the attributes screen. These four linked attributes are edited together. Internally the eDRG used a fifth field DQL History Table but this one is not displayed on any screen.

- ▶ Click [**Edit**].
  ↳ The edit screen opens:



↳ Under the list of DQL Select Queries, a list of available variables is displayed (see paragraph 4.3.4).

The value entered for **DQL Name** can be used in the XSL stylesheet to access the results of the query.

Attribute	Description
DQL Name	The name for the DQL query.
DQL Select query	A valid DQL 'select' query.
Result Threshold	The minimum number of results for this query (default value: 0). See paragraph 4.3.3.
Save Report	Indicates if the results from this query should be saved in

	historic data tables. See paragraph 0.

NOTE

DQL Select Query must be a valid DQL 'select' query. If a query is entered that is not a 'select' query (e.g. 'update' or 'delete'), the report will not be generated.
A DQL query can contain variables (see paragraph 4.3.4) and special functions (see paragraph 4.3.9).

To add new entries:

- ▶ Fill in the fields in the bottom row.
- ▶ Click [**Add**].

To edit an existing entry:

- ▶ Click [**Edit**] for that row.
- ▶ Edit the existing entry.
- ▶ Click [**OK**].

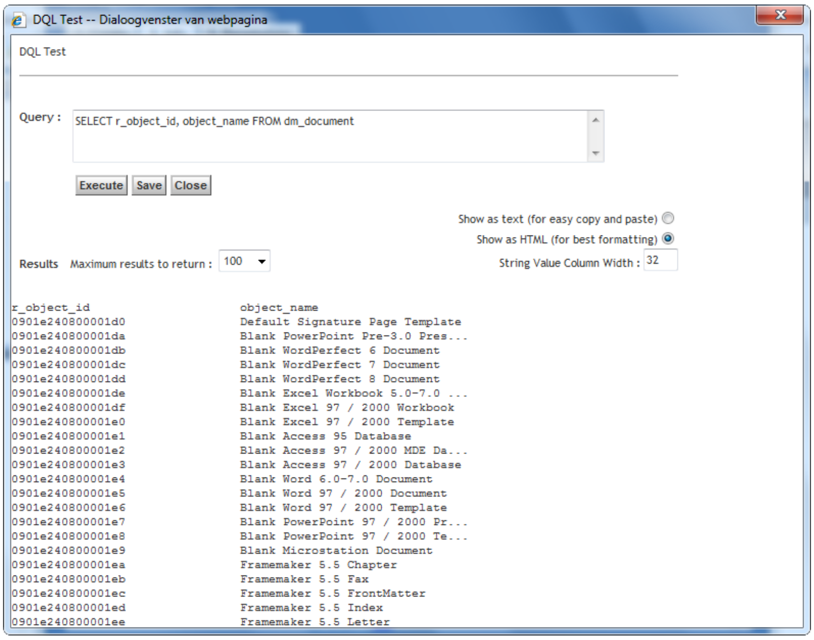
If you select more rows:

- ▶ Move the rows up or down with the [**Move up**] and [**Move down**] buttons.

Or

- ▶ Remove the rows with the [**Remove**] button.
- ▶ Click [**OK**].
  - ↳ All changes are saved.

The [**Test DQL**] button opens DQL Test screen where the user can execute the DQL query and check the results.



Clicking on [**Save**] will copy the DQL query back to the field on the edit page. Clicking on [**Close**] will close the DQL Test screen without changing anything.

NOTE

The DQL Test screen **does not** support variables; variables need to be filled in manually. The DQL Test screen **does** support special functions.

4.3.3 Result Threshold

The Result Thresholds can be used to specify the minimum number of required results. A report is only generated if at least one of the DQL queries for the report has at least the number of results specified at Result Threshold. If you set the threshold for all DQL queries to one, the report will only be generated if at least one of the DQL queries has results. Result thresholds are only checked if the Report Definition is run by a report schedule. Manually generated reports are always generated.

Examples:

Threshold DQL 1	Threshold DQL 2	Actual results DQL 1	Actual results DQL 2	Report generated
0	0	0	0	Yes (both are more or equal)
0	0	5	10	Yes (both are more or equal)
5	5	4	3	No (none are more or equal)
5	5	5	3	Yes (DQL 1 is more or equal)
5	5	4	5	Yes (DQL 2 is more or equal)
5	5	5	10	Yes (both are more or equal)

4.3.4 Variables

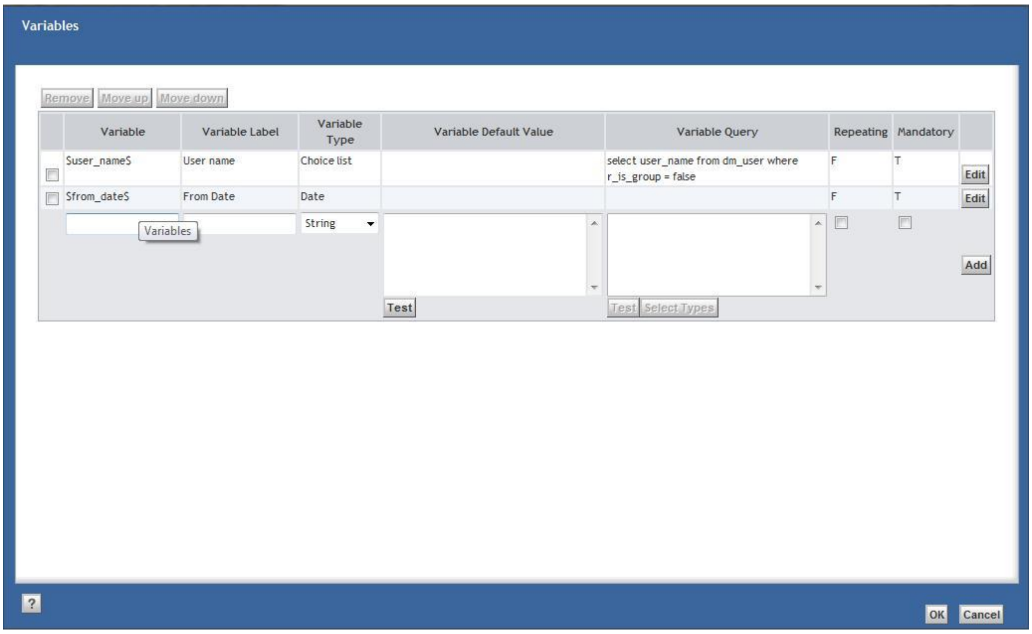
Variables in the DQL query can be used to let the user specify a part of the DQL query that should be run. E.g. the name of a department to get the results for that department. Variables are defined on a Report Definition and consist of the following fields:

Field	Description
Variable	The variable as it is used in the DQL query *).
Variable Label	The label of the variable that is shown to the user running the report definition.
Variable Type	The type of the variable (see next table).
Variable Default Value	The default value for this variable. This value is used as the default for when the user is asked to fill in the variables and is used as the value to use when this Report Definition is run in a schedule. It is possible to use a DQL Select query as default value.
Variable Query	If the Variable Type is set to 'Choice list', this query is used to fill the dropdown list with values the user can choose from.

	If the Variable Type is set to 'Object' then this field lists the allowed object types to add to this variable. This field is ignored for all other variable types.
Repeating	If selected, the user can specify multiple values for this variable. This option is not available for Date or Boolean variables. See paragraph 4.3.6.
Mandatory	If selected then the user has to specify a value for this variable. See paragraph 4.3.7.

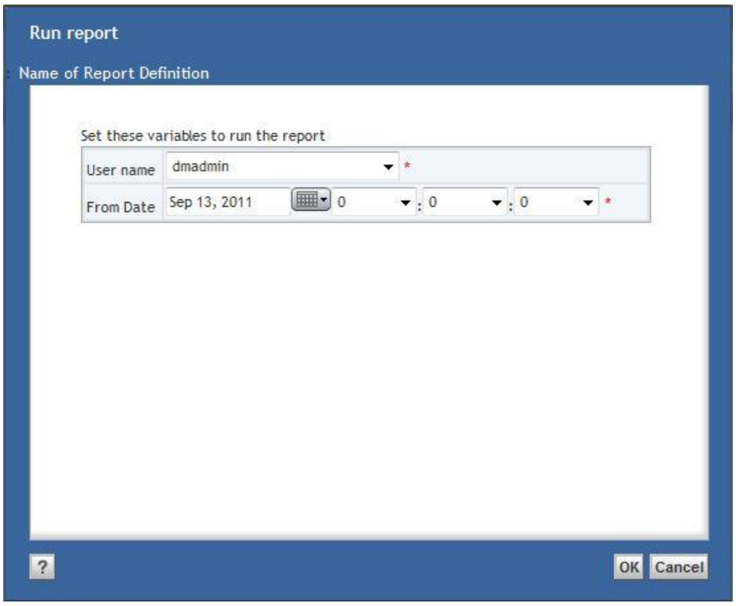
*) There is no required syntax for the variables but it is recommended to use special characters (for example $ characters) in front and at the end of the variable to make them easily recognizable.

Variable Type	Description
String	The user can enter any value for this variable .
Date	The user can choose a date from a date selector.
Integer	The user can enter any positive integer value for this value.
Choice List	The user gets a dropdown list from which it can choose the value. The dropdown list is filled with the values from the DQL query specified in the Variable Query field.
Object	The user can select an object in the repository from an object selector. See paragraph 4.3.5.
Boolean	The user can check a checkbox.



Variables Screen.

To use a variable in a DQL query, you just insert the name of the variable in the DQL query on the place where it should be used. All occurrences of this text are replaced with the value entered by the user.
E.g. if the variables are set as shown in the image, the user that runs the report definition will get the following screen:



If the DQL query for this Report Definition is set to '**SELECT** object_name, r_creation_date **FROM** **dm_document** **WHERE** owner_name='$user_name$' **AND** r_creation_date >= $from_date$ **ORDER BY** r_creation_date', $user_name$ is replaced by the value selected in the first row and $from_date$ is replaced by the date selected on the second row. The result is a list of all documents created by the user from the specified date.

To add new entries:

- Fill in the fields in the bottom row.
- Click [**Add**].

To edit an existing entry:

- Click [**Edit**] for that row.
- Edit the existing entry.
- Click [**OK**].

If you select more rows:

- Move the rows up or down with the [**Move up**] and [**Move down**] buttons.

Or

- Remove the rows with the [**Remove**] button.
- Click [**OK**].
  - All changes are saved.

The [**Test**] button opens the DQL Test screen where the user can execute the DQL query and check the results.

The [**Select Types**] button is only available for object variables; see paragraph 4.3.5.

4.3.5 Object Variables

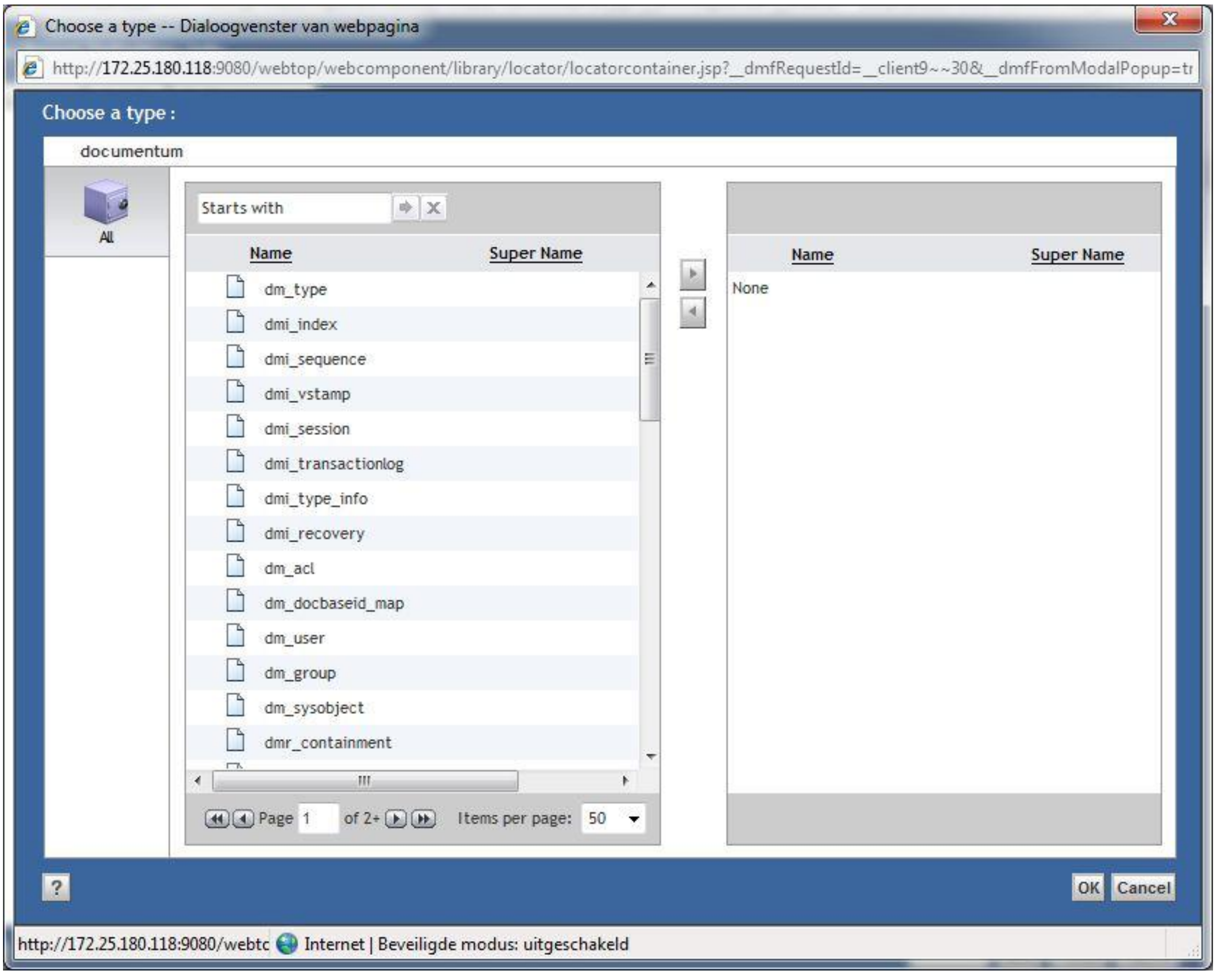
Object Variables allow a user to select an object from the repository as value for a variable. The Object Type of the object the user can select, can be configured.

To use these variables in the DQL query, you can specify the attribute of the object you want to use between brackets.
For example, if your variable is $object$ and you want to use the attribute object_name you specify:
$object$(object_name)
If you don't add an attribute, the object id of the object will be used.

The [**Select Types**] button opens a screen where you can select the allowed object types for the 'Object' variable. All object types are allowed if you don't select an allowed Object Type.

NOTE

The allowed object types are not inherited. For example, if you only select dm_document then dm_document objects are allowed but not subtypes of dm_document.



4.3.6 Repeating Variables

If the Repeating option for a variable is selected, the user can specify multiple values for that variable.

You can use all variable types except dates and booleans as a Repeating Variable.
You can use the Repeating Variable in various locations of you DQL query.

Examples:
In the SELECT part
DQL Query: **SELECT** r_object_id, **var FROM dm_sysobject**

If var is filled with: 'object_name', 'title' and 'object_type' then the actual DQL query will be: **SELECT** r_object_id, **object_name, title, object_type FROM dm_sysobject**

In an IN() or EXISTS () list
DQL Query: **SELECT** r_object_id **FROM dm_sysobject WHERE** object_name **IN** ('**var**')

If var is filled with: 'Example Document' and 'My Document' then the actual DQL query will be: **SELECT** r_object_id **FROM dm_sysobject WHERE** object_name **IN** ('**Example Document**', '**My Document**')
If you don't include the quotes around 'var', there will be no quotes in the actual DQL query.

In a comparison operator
This example is with the '=' operator but it also applies to 'like', 'not like', '<', '>', '!=', '<=', '>=',
DQL Query: **SELECT** r_object_id **FROM dm_sysobject WHERE object_name = 'var'**

If var is filled with 'Example Document' and 'My Document' then the actual DQL query will be: **SELECT** r_object_id **FROM dm_sysobject WHERE (object_name ='Example Document' OR object_name = 'My Document')**
If you don't include the quotes around 'var' , there will be no quotes in the actual DQL query.

If the variable in the DQL query has a '%' character before or after it, this will be included in all the values.

Example:
DQL Query: **SELECT** r_object_id **FROM dm_sysobject WHERE object_name LIKE 'var%'**

If var is filled with 'Example Document' and 'My Document' then the actual DQL query will be: **SELECT** r_object_id **FROM dm_sysobject WHERE (object_name LIKE 'Example Document%' OR object_name LIKE 'My Document%')**

If the statement with the repeating variable is an 'ANY' statement, the ANY statement is repeated as well.

Example:
DQL Query: **SELECT** r_object_id **FROM dm_sysobject WHERE ANY keyword = 'var'**

If var is filled with 'Document' and 'Example' then the actual DQL query will be:
SELECT r_object_id **FROM dm_sysobject WHERE (ANY keyword = 'Document' OR ANY keyword = 'Example')**

NOTE

Repeating variables cannot be used inside special DQL functions. See paragraph 4.3.9.

4.3.7 Optional Variables

The Mandatory field determines if a variable is mandatory or optional. Optional Variables can be left empty by the end user.
Using the special DQL function DRGIF (see paragraph 4.3.9), you can create a DQL select query with optional parts.

Example:
Variable: $owner$
DQL Query: **SELECT** * **FROM dm_sysobject DRGIF**($owner$, **WHERE** owner_name = '$owner$')

If $owner$ is empty the actual DQL query will be:
SELECT * **FROM dm_sysobject**
If $owner$ is dmadmin the actual DQL query will be:
SELECT * **FROM dm_sysobject WHERE** owner_name = 'dmadmin'

4.3.8 Category Variables

Variables can also be defined on a Report Category. In this case all Report Definitions in that Report Category and all sub categories, automatically have these variables.

4.3.9 Special DQL Functions

The euroscript Documentum Report Generator recognizes some additional functions in DQL queries that are not available in normal DQL.
The functions are case-sensitive and should be written in all capitals.

TIP: in many situations eDRG DQL functions can be nested.

DRGIF
This function works like an if-then-else construction. The else part is optional, so the function takes either two or three arguments:
1st: the name of the eDRG variable to check (required), supported values are true and false for boolean variables or an empty string (false) and any other string (true) for other variable types (the literal value false for non-boolean variables also resolves to false).
2nd: the string to return if the first parameter is true.
3rd: the string to return if the first parameter is false. (Optional: if not present, the else part resolves to an empty string).

Examples:
DRGIF(somevar,string1)
DRGIF(somevar,string1,string2)

The string parameters have no delimiters; they contain all characters between commas and braces. They are not trimmed; meaning that you could pass one or more spaces if you wanted to. eDRG variables nested inside string parameters, are resolved.

NOTE

If you want to use commas inside string parameters, you must escape them by specifying two commas.

NOTE

If you use variables inside a DRGIF function, the enter value for the variable cannot contain a single comma. They too need two commas.

DRGINMONTH
This function tests if the specified attribute is in one or more months. The function takes three arguments:
1st: the name of the attribute to check (required).
2nd: the month offset. 0 = this month, -1 = previous month, -2 = 2 months ago etc. (Optional; if not specified, 0 is used (current month)).
3rd: the number of months to include from the offset. (Optional; if not specified, 1 is used).

Examples:
DRGINMONTH(r_creation_date): r_creation_date must be in the current month.
DRGINMONTH(r_creation_date, -1): r_creation_date must be in the previous month.
DRGINMONTH(r_creation_date, -1, 2): r_creation_date must be in the previous month or in this month.

DRGINWEEK
This function tests if the specified attribute is in one or more weeks. The function takes three arguments:
1st: the name of the attribute to check (required).
2nd: the week offset. 0 = this week, -1 = previous week, -2 = 2 week ago etc. (Optional; if not specified, 0 is used (current week)).
3rd: the number of weeks to include from the offset. (Optional; if not specified, 1 is used).

Examples:
DRGINWEEK(r_creation_date): r_creation_date must be in the current week.
DRGINWEEK(r_creation_date, -1): r_creation_date must be in the previous week.
DRGINWEEK(r_creation_date, -1, 2): r_creation_date must be in the previous week or in this week.

DRGINDAY
This function tests if the specified attribute is in one or more days. The function takes three arguments:
1st: the name of the attribute to check (required).
2nd: the day offset. 0 = today, -1 = yesterday, -2 = 2 days ago etc. (Optional; if not specified, 0 is used (today)).
3rd: the number of days to include from the offset. (Optional; if not specified, 1 is used).

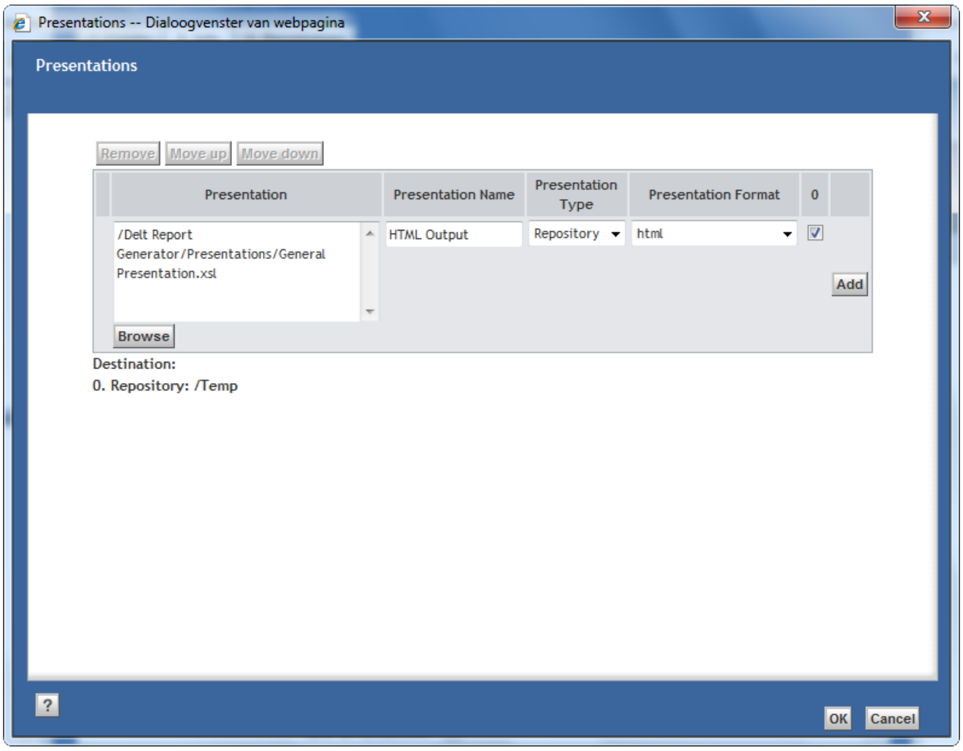
Examples:
DRGINDAY(r_creation_date): r_creation_date must be in the current day.
DRGINDAY(r_creation_date, -1): r_creation_date must be in the previous day.
DRGINDAY(r_creation_date, -1, 2): r_creation_date must be in the previous day or in this day.

4.3.10 Presentations

Presentations define the stylesheets that are used to create the reports. Each Presentation consists of a presentation specification; a Presentation Name, a Presentation Type, a Presentation Format and a Presentation Destination specification.
Only the attribute Presentation is displayed on the attributes screen. These five linked attributes are edited together. A destination specification allows selection of one or more previously defined destinations.

> **NOTE**
>
> You must have created at least 1 destination before you can create presentations.

- Click [**Edit**].
  - The edit screen opens:



- **Presentation** is the full location and name of the XSL presentation file.
- **Presentation Name** is the name of the presentation that will be shown to the user generating the report, if multiple presentations are specified for this report definition
- **Presentation Type** specifies if the presentation file is on the file system (OS) or in the repository folder structure.
- **Presentation Format** specifies in which format the result will be saved (this should correspond to the output of the XSL file).
- **Presentation Destination** The numbers 0 to … represent the available destinations.
  The list of defined destinations is displayed at the bottom of the page.

NOTE

Manually generated reports are always placed in the user's home cabinet, regardless of any destinations specified.

The [**Browse**] button lets the user select a stylesheet document in the repository as presentation file. This button only works if Presentation Type is set to Repository. If the Presentation Type is set to any other value, this button will be disabled.
If you use the [**Browse**] button to select a presentation file of the object type Report Presentation (see paragraph 4.8), the Presentation Type is automatically set correctly.

4.3.11 Historic Data

NOTE

Historic data can be globally disabled. If it is disabled, you are still able to enable historic data on Report Definitions. However, these settings are ignored.

NOTE

If you enable historic data on a Report Definition, you also need to make sure the Report Presentation can process the historic data.

Historic data can be enabled on Report Definitions. This means that results from the queries are stored in the database and included in future reports.

Set the **Save Report** attribute to true for each DQL Select Query for which historic data should be stored. It is possible to enable this for only some of the DQL Select Queries of a Report Definition.
You can control how often historic data is stored (e.g. at most once a week) by setting the below attributes on the Report Definition.

Attribute	Description	Values
Save Run Mode	This attribute controls if historic is stored for only automatic runs (schedule), only manual runs or both.	• Automatic • Manually • Automatic and Manually
Save Report Interval	These attributes determine the minimum time required between 2 saved reports. E.g. if this is set to 1 hour and a report is saved at 13:00:00 then all reports generated between 13:00:00 and 14:00:00 will not be stored.	
Save Report Mode		• Minutes • Hours • Days • Weeks • Months • Years

You can also control how long the historic data is stored by setting the following attributes on the Report Definition. A job runs periodically to remove the expired data.

Attribute	Description	Values
Retention Interval	These attributes determine the maximum time reports will be kept in the database.	
Retention Mode		• Days • Weeks • Months • Years

Use the last three attributes to determine the default settings for including historic data in a report.

Attribute	Description	Values
Default Show History	Historic data is included by default if checked.	
Default Show History Interval	These attributes determine the default period of historic data to be included.	
Default Show History Period		• Last x Days • Last x Weeks • Last x Months • Last x Years

NOTE

The PDF format of the demo report "Number of Documents by Object Type" can handle historic data. Storing the historic data for this report is not enabled by default. Check the "Save Report" option to enable storing the historic data.

4.3.12 Permissions

On the **Permissions tab** of a Report Definition, the permissions can be changed.

NOTE

By default each report definition has the dlt_report_definition_acl ACL (unless the Report Category has a different ACL), which means:

- Users with the **report_user** role can see the report definition and can generate reports.
- Users with the **report_builder** role also can edit the report definition.
- Users with the **report_administrator** role also can delete the report definition.

4.4 Automatically Removing Old Reports

Generated reports can be automatically deleted by setting the attributes 'Remove Reports after Number of Versions' and/or 'Remove Reports after Number of Days' on the report definition.

The 'Remove Reports after Number of Versions' option removes the oldest versions of the object if the number of versions exceed the specified number (0 means do not remove reports) when a new version is added.
The 'Remove Reports after Number of Days' options removes the versions of a report older than the specified number of days (0 means do not remove reports) when a new version is added.

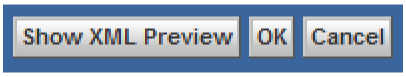
Limitations for this function:
Automatically removing reports only works on reports that are stored in the repository.
Automatically removing reports only works if 'Save as Version' is selected.
Automatically removing reports only works on the current report name. E.g. if the report name is changed on the report definition, the reports generated with the old name will not be removed.
If two report definitions both use the same destination and report name for the generated reports, reports generated by both Report Definitions count towards the total number of versions.

4.5 Getting the XML File

CAUTION

Make sure that all changes are saved before you click the [**Show XML Preview**] button, otherwise the old settings will be used.

Report Builders have an additional button on the properties page of report definitions. This button can be used to get the XML file that is processed by the style sheet. This XML file can then be used to test the stylesheets without having to import it and run the report definition every time.



- The generated XML file is placed in the user's home cabinet and automatically opened.

4.6 Copying/Moving Report Definitions

Copying and moving Report Definitions works in the same way as copying or moving other objects.
It is only possible to copy or move Report Definitions to Report Categories from **Report Administration\Report Definitions**.

4.7 Exporting/Importing Report Definitions

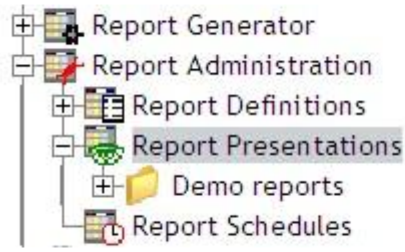
Report Definitions can be exported by Report Builders:
Navigate to **Report Administration\Report Definitions**.
Select the report definition(s) that you want to export.
Choose menu item **File**.
Choose the submenu item **Export**.
Select the location where you want to save the exported files.

Report definitions can be imported by report builders:
Navigate to **Report Administration\Report Definitions**.

Navigate to the report category where the report definition should be imported.
Choose menu item **File**.
Choose the submenu item **Import**.
Select the file you want to import.

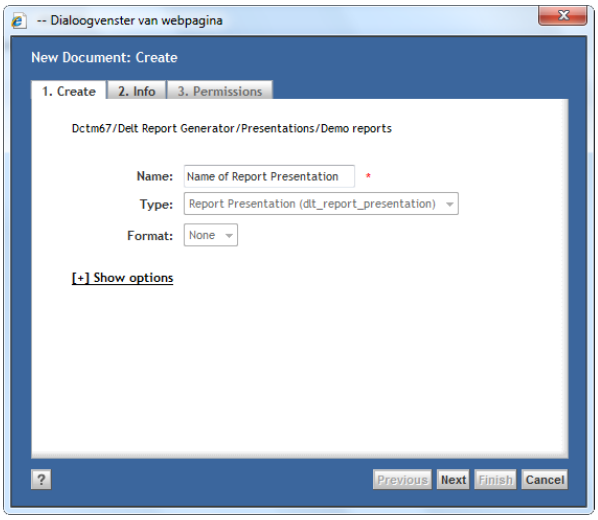
4.8 Creating Presentation Files

The stylesheets use XSL (http://www.w3.org/TR/xsl) to create the reports.
The Presentation Files can be accessed from the tree node **Report Administration\Report Presentations**.



To create a report presentation:

- Navigate to the folder in which you want to create the report presentation.
- Choose the menu item **File**.
- Choose the submenu item **New**.
- Choose the submenu item **Report Presentation**.
  - The following screen is displayed:

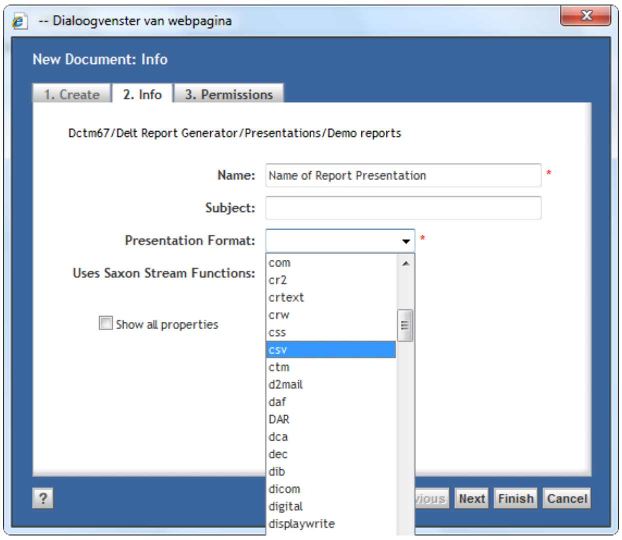


- Fill in a name for the Report Definition.

NOTE

The report presentation name is subject to the regular object_name attribute restrictions; e.g. has a maximum length of 255 characters.

The attributes shown on the info tab, define the Report Presentation.



- Fill in the required attributes.
- Click [**Finish**].
  - ↳ The Report Definition is created.

Explanation of the attributes:

Attribute	Description	Limits
Name	Name of the report presentation.	255 characters
Subject	Subject of the report presentation.	192 characters
Presentation Format	Specifies in which format the result will be saved (this should correspond to the output of the XSL file). This setting is always used regardless of the format that is set for this presentation file on the report definition.	
Uses Saxon Stream Functions	Indicates if the presentation uses the Saxon Stream function to access the XML result. See paragraph 4.8.5.	

4.8.1 Including Other Presentation Files

It is possible to use xsl:include and xsl:import to include other stylesheets into your own stylesheet. Paths used for xsl:include and xsl:import can be an absolute path or a relative

path to the style sheet that includes them. A stylesheet in the repository can only include other stylesheets in the repository and not stylesheets that are on the OS file system.
A stylesheet on the OS file system can only include other stylesheets on the OS file system and not stylesheets that are in the repository.

4.8.2 Creating Presentation Files for PDF Report

PDF reports can be created by using XSL-FO (http://www.w3.org/TR/xsl/#fo-section). The stylesheet that is used by the Report Definition should create an XSL-FO file which is then transformed to a PDF file.

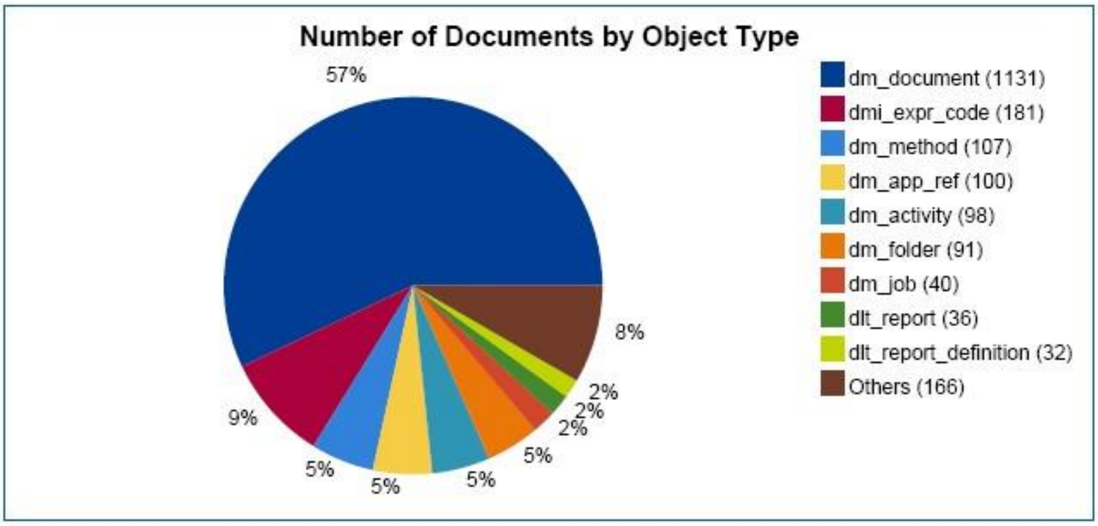
NOTE

The transformation from XSL-FO to PDF will only happen if the presentation type for this presentation file (on the report definition) is set to PDF.

The euroscript Documentum Report Generator uses Apache FOP (http://xmlgraphics.apache.org/fop) to create the PDF files. The limitations of Apache FOP also apply to the euroscript Documentum Report Generator.

4.8.3 Charts in PDF Reports

Charts can be created by generating SVG images (http://www.w3.org/TR/SVG11) inside the XSL-FO file.



Example chart that can be created in PDF files.

When the demo reports are installed, some of the examples create PDF files and one of them (charts.xsl) can create charts. This stylesheet can be included in other stylesheets that create PDF files. Read the comments in the charts.xsl file for instructions on how to use it.

Apache FOP uses Batik (http://xmlgraphics.apache.org/batik) to generate the SVG images. The limitations of Batik also apply to the euroscript Documentum Report Generator.

4.8.4 Logging Errors in XSL-FO

Because of the logging method used by Apache FOP and Batik to log errors, we cannot place those messages in the euroscript Documentum Report Generator log file.
The post installation tasks describe a workaround, so the messages will be logged to a separate log file on the content server host.

4.8.5 Process Large XML Files with Saxon Stream Function

Processing large XML files requires lots of memory on the Java Method Server. If your reports are too big to process in memory, you can use the Saxon Stream Function to reduce the memory requirements.

To use the Saxon Stream function, you need to have Saxon-EE (Saxon-SA) installed and a valid license to use it. Software and licenses can be obtained via the Saxonica website http://www.saxonica.com

Select the 'Uses Saxon Stream Functions' attribute on the report presentation object to enable the Saxon Stream function for that report presentation.

With this option enabled, the euroscript Documentum Report Generator will create 2 XML files. 1 file is the same as the XML file for all reports (see paragraph 4.9). The other file looks like this:

```
<?input-file filename="*Path to actual result file*"?>
<dummy/>
```

The second file is passed through to the presentation file. This presentation file can use the following two lines to load the actual XML result file and process it.

```
<xsl:param name="SourceDocument" select="/processing-instruction('input-file')/saxon:get-pseudo-attribute('filename')" />

<xsl:apply-templates
select="saxon:stream(doc($SourceDocument)/result/objects)" />
```

4.9 XML File

The basis syntax of XML file that the style sheet should process, is:

```
<result>
   <settings>
      <variable name="*name of first variable*">*value first variable*</variable>
      …
      <variable name="*name of last variable*">*value last variable*</variable>
      <format type="float">*float format*</ format>
      <format type="date">*date format*</ format>
      <reportname>*name of the report*</reportname>
   </settings>
   <objects name="*name of first DQL query*">
      <object>
         <*first column name* type="*type*">*value first row*</*first column name*>
```

```
            …
            <*last column name* type="*type*">*value first row*</*last column name*>
        </object>
        …
        <object>
            <*first column name* type="*type*">*value last row*</*first column name*>
            …
            <*last column name* type="*type*">*value last row*</*last column name*>
        </object>
    </objects>
    …
    <objects name="*name of last DQL query*">
        <object>
            <*first column name* type="*type*">*value first row*</*first column name*>
            …
            <*last column name* type="*type*">*value first row*</*last column name*>
        </object>
        …
        <object>
            <*first column name* type="*type*">*value last row*</*first column name*>
            …
            <*last column name* type="*type*">*value last row*</*last column name*>
        </object>
    </objects>
    <query_times>
        <duration>*duration of the first DQL query in milliseconds*</duration>
        …
        <duration>*duration of the last DQL query in milliseconds*</duration>
    </query_times>
</result>
```

The settings part of the XML file contains the values set by the user for the variables (if any were specified for this Report Definition) and the float and date formats set on the report definition by the Report Builder.

The available column types are:

- boolean
- double
- id
- integer
- string
- time
- undefined

4.9.1 Values From Repeating Attributes in The XML File

The syntax of values from repeating attributes in the XML file depends on the value of the 'Split Repeating Values' attribute on the report definition.

If the 'Split Repeating Values' attribute is not selected, the syntax is:

```
<column name* type="*type*">*value1*,*value2*,...,*last value*</*column name*>
```

If the 'Split Repeating Values' attribute is selected, the syntax is:

```
<column name* type="*type*" repeating="true">
        <value index="0">*value1*</value>
        <value index="1">*value2*</value>
        ...
        <value index="...">* last value*</value>
</*column name*>
```

4.9.2 Historic Data in The XML Files

If the user requested historic data to be included in the report, the following XML syntax is added to the report after </settings> and the first <objects>.

```
   <historic name="*name of first DQL query*">
      <historic_report reporttime="*date of first included historic data*"
reportuser="*useraccount used for this report*" runtype="*1 = manual run, 2 = automatic
run*">
         <historic_object>
            <*first column name* type="*type*">*value first row*</*first column name*>
            ...
            <*last column name* type="*type*">*value first row*</*last column name*>
         </ historic_object>
         ...
         <historic_object>
            <*first column name* type="*type*">*value last row*</*first column name*>
            ...
            <*last column name* type="*type*">*value last row*</*last column name*>
         </ historic_object>
      </historic_report>
      ...
      <historic_report reporttime="*date of last included historic data*"
reportuser="*useraccount used for this report*" runtype="*1 = manual run, 2 = automatic
run*">
         <historic_object>
            <*first column name* type="*type*">*value first row*</*first column name*>
            ...
            <*last column name* type="*type*">*value first row*</*last column name*>
         </ historic_object>
         ...
         <historic_object>
            <*first column name* type="*type*">*value last row*</*first column name*>
            ...
            <*last column name* type="*type*">*value last row*</*last column name*>
         </ historic_object>
      </historic_report>
   </historic>
   ...
   <historic name="*name of last DQL query*">
```

</historic>

DQL Queries with historic data within the selected period are added to the XML like the "first DQL query" example. DQL Queries with historic data disabled or no historic data within the selected period, are added to the XML like the "last DQL query" example.

If historic data is globally disabled, no <historic> tags will be added to the XML.

4.10 eDRG usage

Usage information about all the generated reports is stored within the repository and can be queried with DQL.

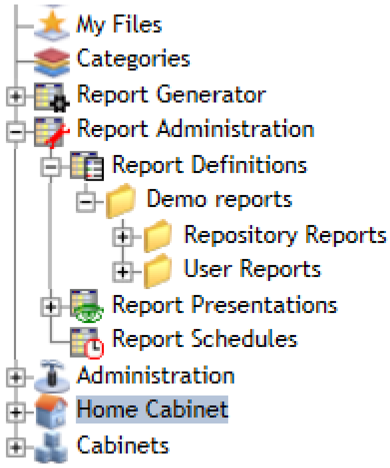
Usage information about schedules is stored in the type edrg_usage_schedule:

Attribute	Type	Description
report_schedule_id	ID	The object ID of the schedule that was executed
start_date	Time	The start date of the schedule
duration	Integer	duration of the schedule in milliseconds

Usage information about report definitions is stored in the type edrg_usage_report

Attribute	Type	Description
report_definition_id	ID	The object ID of the report definition that was executed
run_user	String	The user that started the report
start_date	Time	The start date of the report definition
duration	Integer	duration of the report in milliseconds
run_type	Integer	1 = manual report, 2 = scheduled report, 3 = test report (by report builder)
schedule_usage_id	ID	object ID of the edrg_usage_schedule object if this report was part of a scheduled run
variable_name	Repeating string	List of variables of the report
variable_value	Repeating string	List of variable values of the report (corresponds with the items in variable_name)
historic_from	Time	The from date selected for historic information (or null for no historic data)
historic_to	Time	The to date selected for historic information (or null for no historic data)
query_duration	Repeating Integer	For each DQL Select query in the report the time it took to run the query in miliseconds
presentation_duration	Repeating Integer	For each presentation in the report the time it took to create the output file in miliseconds

5 Report Administrator



5.1 Introduction

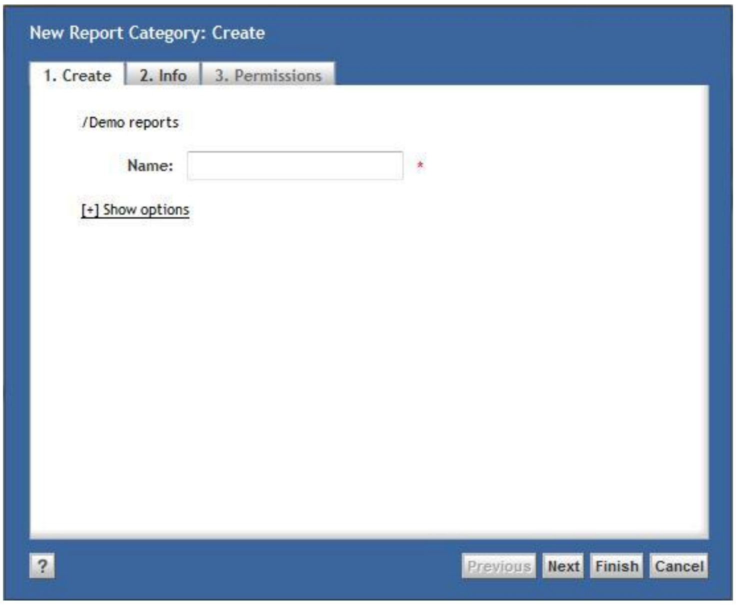
This chapter describes the functionality for users with the report_administrator role.
Report Administrators can access the "Report Schedules" tree node under the "Report Administration" tree node.
The "Report Schedules" tree node contains the Report Schedules for the eDRG.

5.2 Creating Report Categories

Report Categories can be created from the following section:
Report Administration\Report Definitions (not from the Report Generator section).

To create a Report Category:

- Navigate to the report category in which you want to create the Report Category.
- Choose menu item **File**.
- Select submenu item **New**.
- Select submenu item **Report Category**.



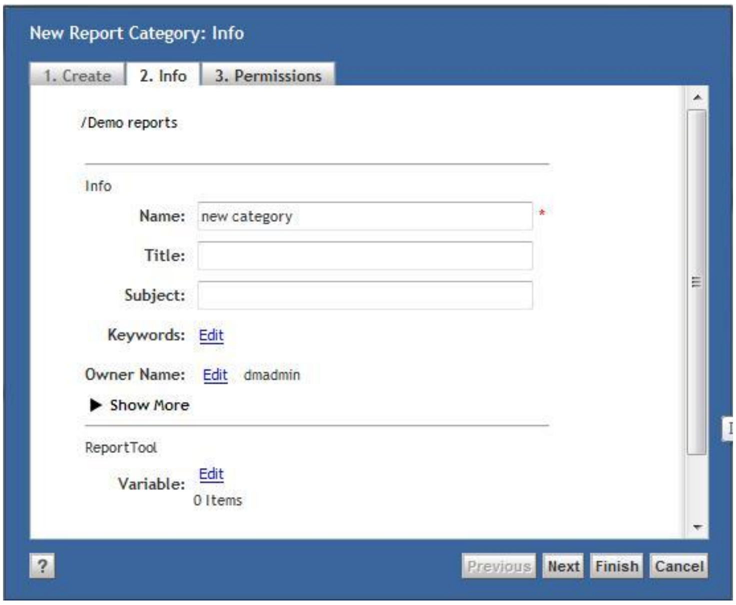
↳ A window opens where you can enter a name for the Report Category.

- Fill in a name for the Report Category.

NOTE

The Report Category Name is subject to the regular object_name attribute restrictions, e.g. has a maximum length of 255 characters.

- Click [**Next**].



↳ The attributes screen for Report Categories is shown.

- Optionally fill in the attributes.
- Click [**Finish**].
  ↳ The Report Definition is created.

Explanation of the attributes:

Attribute	Description	Limits
Variable *)	The variables for the reports in this category. See paragraph 4.3.4	32 characters
Variable Label 1)		255 characters
Variable Type 1) *)		
Variable Default Value 1)		255 characters
Variable Query 1)		1000 characters
Variable Repeating 1)		
Variable Mandatory		

1) These attributes are hidden and won't be displayed on the attributes page but can be set by clicking on the edit link for the visible attribute in the same group (i.e. for variables the attribute variable is visible and the attribute variable_type is not)

*) These attributes are required

5.2.1 Permissions

On the Permissions tab of a Report Category, the permissions can be changed.
By default each Report Category will have the dlt_report_definition_acl ACL, which means:

- Users with the **report_user role** can see the Report Category.
- Users with the **report_builder role** can edit the report category and add Report Definitions.
- Users with the **report_administrator role** can delete Report Category.

NOTE

If you change the permissions on a Report Category, the new permissions will be inherited by all new Report Category and Report Definitions that are created in this Report Category (not for the objects that are already present).

5.3 Deleting Report Categories

Report Categories can be deleted:

- ▶ Navigate to Report Administration\Report Definitions.
- ▶ Select the report category that you want to delete.
- ▶ Choose menu item **File**.
- ▶ Choose submenu item **Delete**.
  - ↳ The Report Category is deleted.

CAUTION

If the Report Category is not empty, all Report Categories and Report Definitions in the Report Category will be deleted as well.

5.4 Deleting Report Definitions

CAUTION

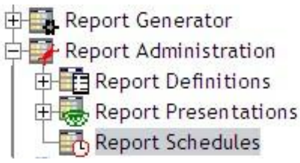
Ensure that the Report Definition is not used in a report schedule before you delete it.

Report Definitions can be deleted:

- ▶ Navigate to Report Administration\Report Definitions.
- ▶ Select the Report Definition that you want to delete.
- ▶ Choose menu item **File**.
- ▶ Choose submenu item **Delete**.

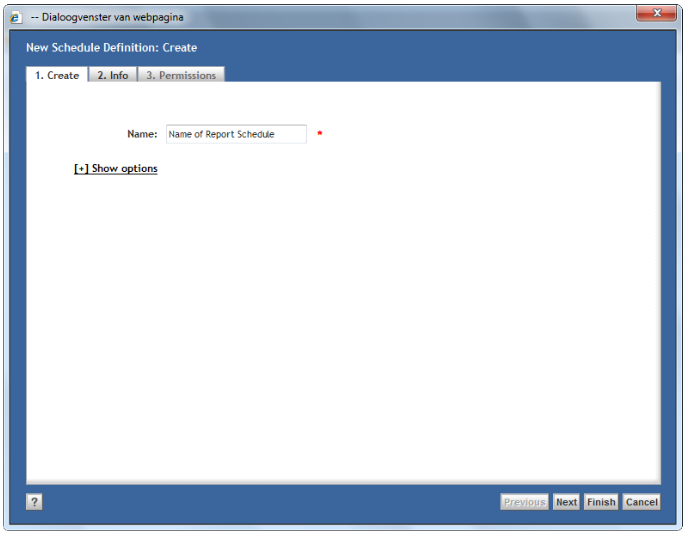
5.5 Creating Report Schedules

Report Schedules can be created by Report Administrators from the following section: **Report Administration\Report Schedules**.



To create a Report Schedule:

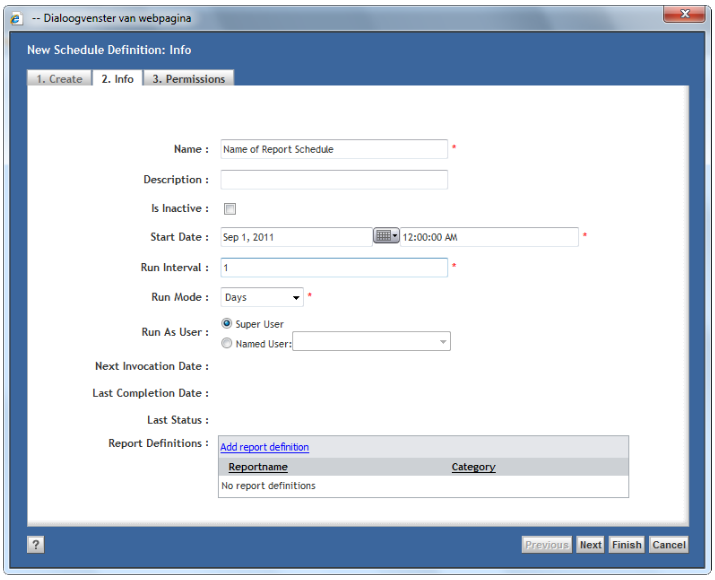
- Navigate to **Report Administration\Report Schedules**.
- Choose menu item **File**.
- Choose submenu item **New**.
- Choose submenu item **Report Schedule**.
  - The following screen opens:



- Fill in a name for the Report Schedule and click Next.
  - The following screen opens:

NOTE

The Report Schedule Name is subject to the regular object_name attribute restrictions; e.g.. has a maximum length of 255 characters.



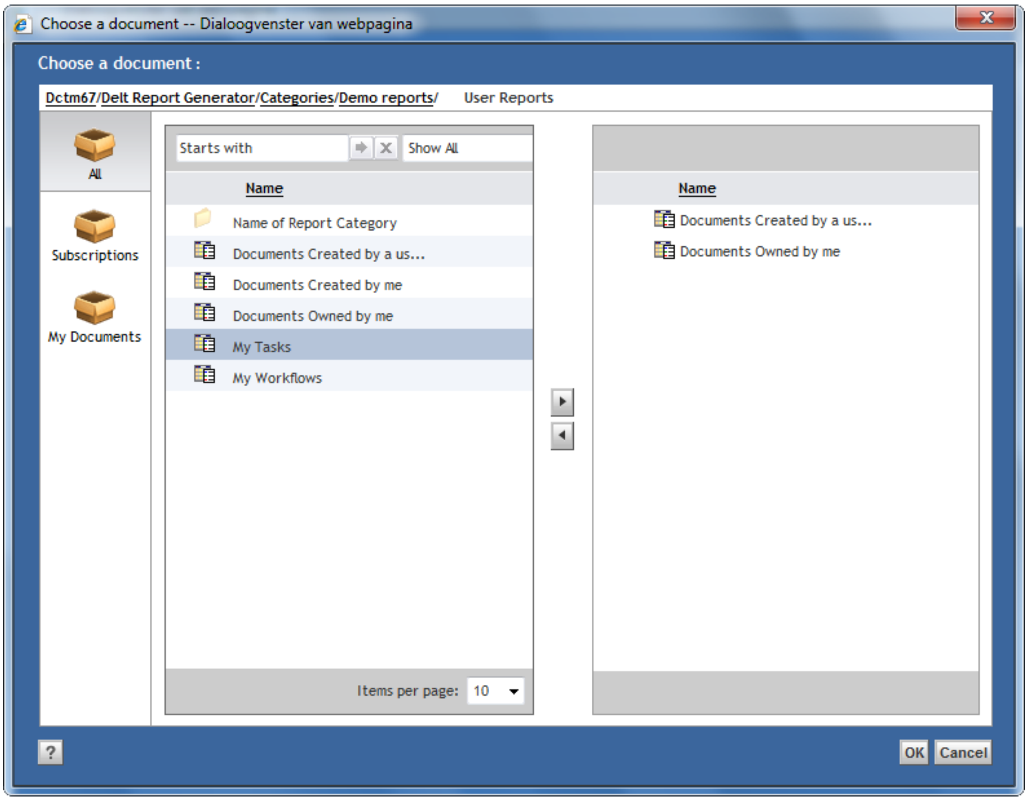
- Fill in the required attributes.

Explanation of the attributes

Attribute	Description	Limits
Name	The name of the report schedule.	255 characters
Description	The description of the report schedule.	255 characters
Is Inactive	If checked, this schedule will be ignored by the Report Generator and not be run.	
Start Date	The first date at which the schedule should run. The next invocation is calculated from this date.	
Run Interval	Used in conjunction with run_mode to determine how often to run the report. For example; if you enter 3 for this attribute and Weeks for run_mode, the job is invoked every three weeks. The default is zero (0) which means the schedule will not run.	
Run Mode	Used in conjunction with run_interval to determine how often to run the report. Run_mode specifies the unit of measure for the value in run_interval.	
Run As User	The schedule will be run under the selected user. This means that the DQL queries only return objects that this user can see. Default setting is Super User.	
Next Invocation Date	The next time the schedule will be run. This value is recalculated every time you save the schedule or the schedule is run.	
Last Completion Date	The last time the schedule was run.	
Last Status	The status message from the last run.	
Report definitions	The list of report definitions added to this schedule.	

To add Report Definitions to this schedule:

- Click on the link **Select Report Definition**.
  - The following screen opens:



- Click [**OK**].
- Click [**Finish**].
  - The Report Schedule is created.

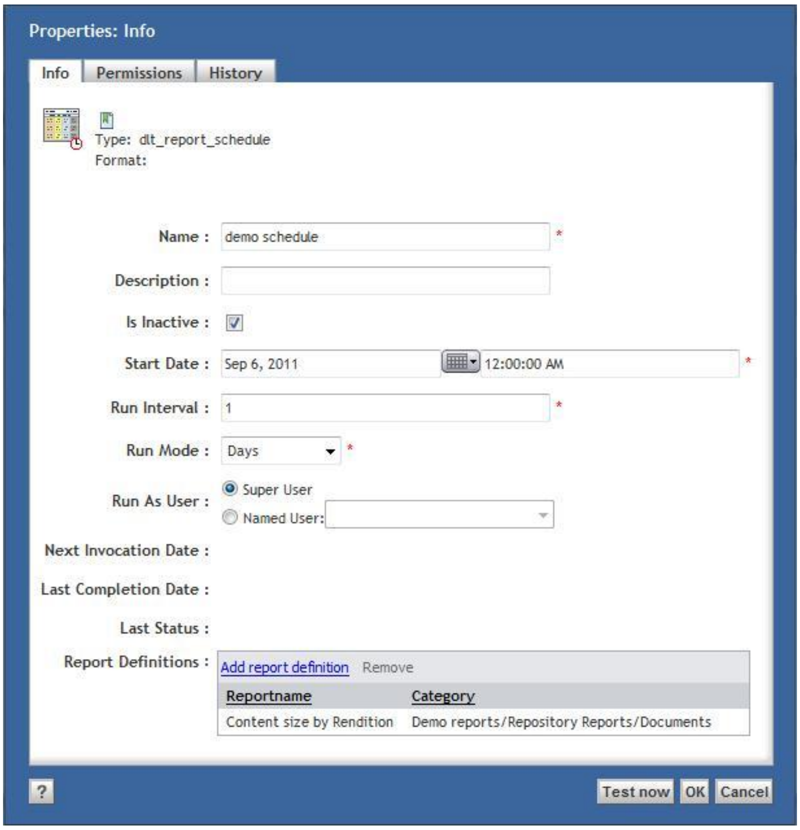
5.6 Testing Report Schedules

CAUTION

Make sure that all changes are saved before you click the [**Test now**] button, otherwise the old settings will be used.

To test a Report Schedule:

- Navigate to the properties of the Report Schedule.



- Click [**Test now**].
  ↳ The schedule runs immediately (without sending a message).

5.7 Exporting/Importing Report Schedules

Report Schedules can be exported:
Navigate to **Report Administration\Report Schedules**.
Select the Report Schedules(s) that you want to export.
Choose menu item **File**.
Choose the submenu item **Export**.
Select the location where you want to save the exported files.

Report Schedules can be imported:
Navigate to **Report Administration\Report Schedules**.
Choose menu item **File**.
Choose the submenu item **Import**.
Select the file you want to import.

5.8 Deleting Report Schedules

Report Schedules can be deleted:

- Navigate to Report Administration\Report Schedules.
- Select the Report Schedule that you want to delete.
- Choose menu item **File**.
- Choose menu item **Delete**.

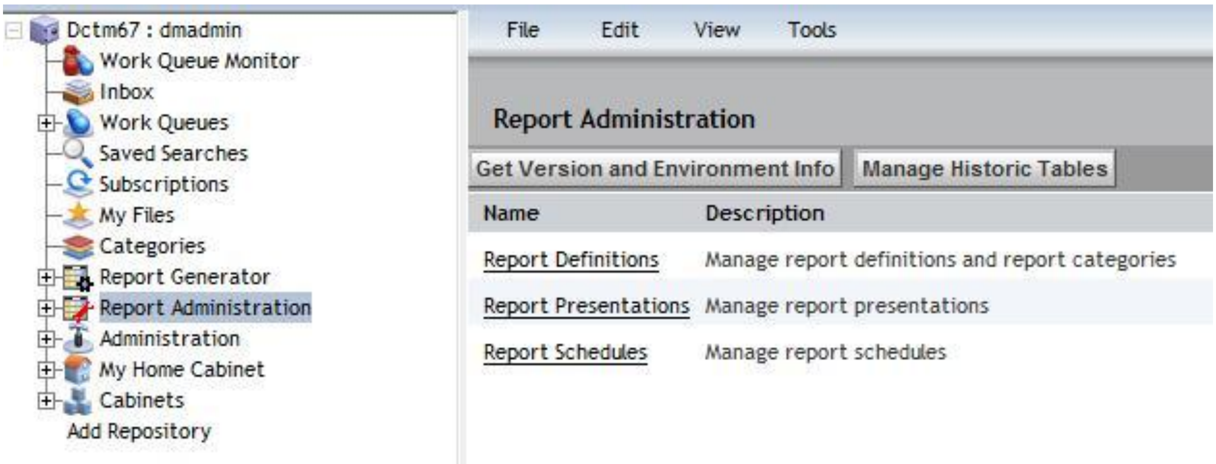
↳ The Report Schedule is deleted.

5.9 Automatically Running Report Schedules

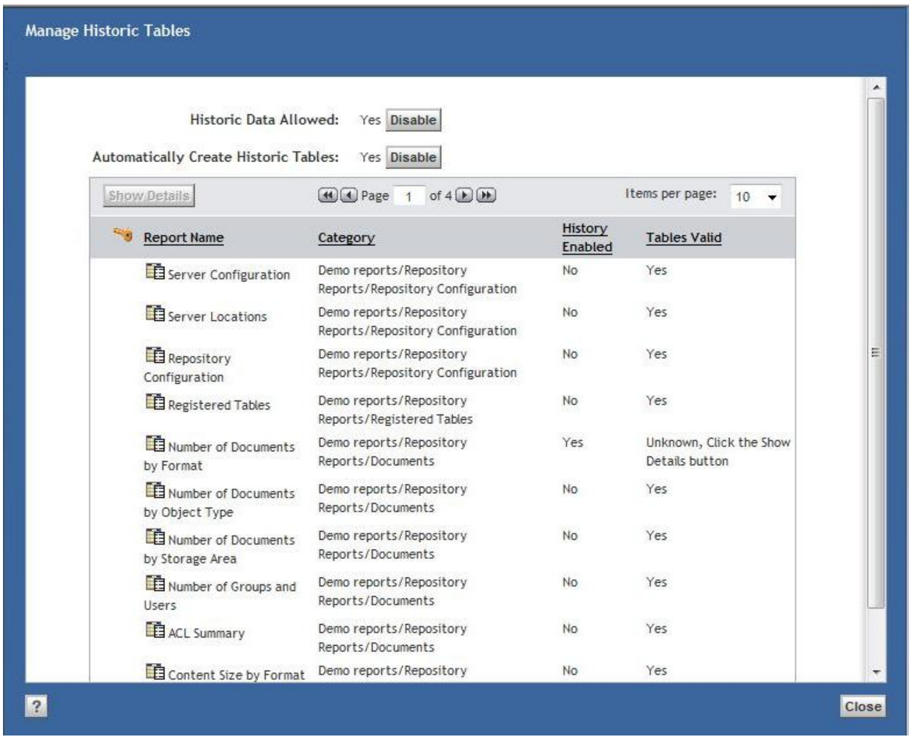
The ‘Delt Reporting Agent’ job periodically checks which schedules should be started and starts them. When a Report Schedule is started, a flag is set so that the schedule won’t get started twice.
If the Documentum Java Method Server or the Documentum Content Server is stopped while a schedule is running, this flag will not be removed. To prevent that those schedules will never run again, the flag is automatically removed after 1 hour.

5.10 Managing Historic Tables

A Report Administrator can manage the historic tables by pressing the **Manage Historic Tables** button on the **Report Administration** page.



This will display the Manage Historic Tables screen.



Historic Data Allowed: indicates if historic data is globally enabled or not. If you are a member of the admingroup, you can enable or disable this setting from this page.

Automatically Create Historic Tables: indicates if historic tables can be automatically created/updated by the eDRG if needed. If you are a member of the admingroup, you can enable or disable this setting from this page.

The table contains all the Report Definitions in the repository.
History Enabled: indicates if historic data is enabled for this report
Tables Valid: indicates if the historic tables are valid.

NOTE

For Report Definitions with historic data enabled, it is not possible to easily detect if the tables are correct. For these the **Tabled Valid** column will be set to **Unknown, Click the Show Details button**.

Select the Report Definition and click the [**Show Details**] button for more information.
This information is displayed in a table under the Report Definitions table.
If the tables with the Report Definition are incorrect, you will get a link to a logfile which contains the problems for that Report Definition

Possible problems are:

Problem	Solution
The table ".." exists for DQL .. but isn't used	Remove the table and registered table from the database.
No name is specified for the historic tables for DQL ..	Nothing exists yet for this Report Definition. Set the dql_history_table to the name of the table you want to use (see paragraph 5.10.1) and create the table and registered table in the database.
The historic table ".." for DQL .. has an invalid prefix. Prefix must be edrg_	Update the dql_history_table field with the correct prefix and update the table and registered table in the database.
The table ".." does not exist but is required for DQL ..	Create the table and registered table in the database.
The column ".." in table ".." for DQL .. is of the type .. but should be ..	Change the column in the table and registered table in the database to the correct type.
The length of the column ".." in table ".." for DQL .. is .. but should be ..	Change the column in the table and registered table in the database to the correct length.
The column ".." in table ".." for DQL .. exists but isn't used	Remove the column from the table and registered table in the database.
The column ".." of the type .. and length .. is missing in the table ".." for DQL ..	Add the column to the table and registered table in the database.

5.10.1 Historic Data Table

Each DQL Select Query with **Save Report** enabled, has a single values Historic Data Table and optionally a repeating values historic data table.
The name of the Single Values Table is the value of the dql_history_table field + **_s**.
The name of the Repeating Values Table is the value of the dql_history_table field + **_r**.

The value in dql_history_table must start with **edrg_**

If tables are automatically generated, they will get the name **edrg_** + object id of the report definition + _ + unique number (6 characters) + **_s** or **_r**

The following table describes the columns that are added to all the Historic Data Tables:

Column	Description
edrg_report_identifier (integer)	Used to group results from a report.
edrg_row_id (integer)	Used to sort the results in a report.
edrg_value_index (integer)	Only added to the repeating values table. Used to sort the repeating values results in a row.

NOTE

The eDRG uses the registered tables from Documentum to check the Historic Data Tables. The information in the registered tables must always match the actual Historic Data Table.

5.10.2 Historic Info Table

The historic data function needs a Historic Info Table to store information about the stored reports. This table needs to be created and registered manually because its automatic creation is disabled and this table does not exist yet (no reports with historic data enabled have previously been run).

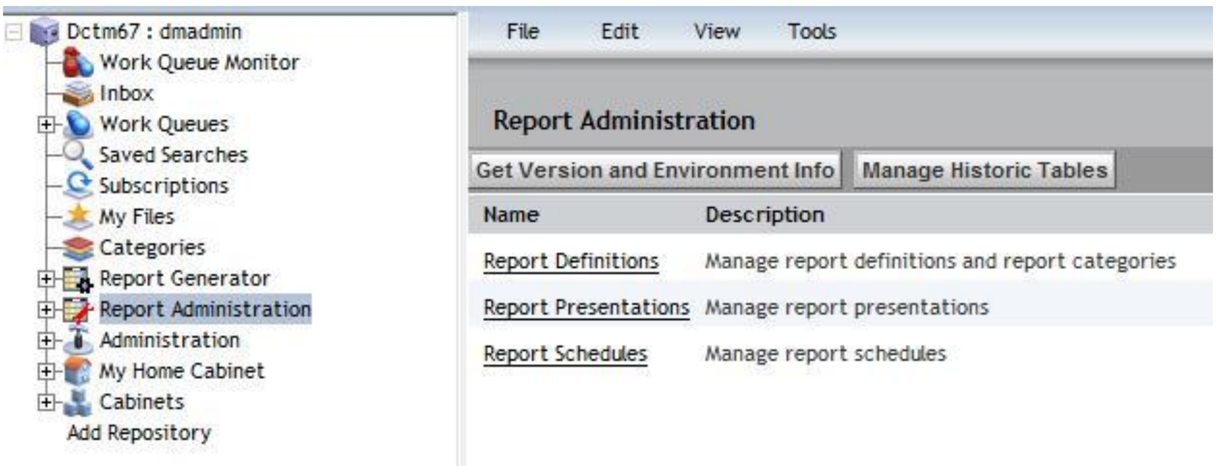
The name of the table is **edrg_history_info** and must contain the following columns:

Column	Type
edrg_report_definition_id	VARCHAR(16)
edrg_report_identifier	INTEGER
report_time	DATETIME
report_user	VARCHAR(32)
run_type	INTEGER
status	INTEGER

Use the following DQL query to register this table:
REGISTER TABLE dm_dbo.edrg_history_info(edrg_report_definition_id **string**(16), edrg_report_identifier **integer**, report_time **date**, report_user **string**(32), run_type **integer**, status **integer**)

5.11 Getting Version and Environment Information

A Report Administrator can get version and environment information about the euroscript Documentum Report Generator and the systems it is running on, by pressing on the [**Get Version and Environment Info**] button on the **Report Administration** page. This will gather the information and display the results.



A result will look like this:

Content Server Information

Content Server Version:	6.7.0000.0217 Win32.Oracle
Delt Report Generator DocApp Version:	3.0

Java Method Server Information

DFC Version:	6.7.0000.0213
Java Version:	Sun Microsystems Inc. 1.6.0_17
Java Total Memory:	265486336
Java Free Memory:	90891304
Operating System:	Windows Server 2008 6.0
Architecture:	x86
Method Server runs as:	dmadmin
XSLT Transformer:	Xalan Java 2.7.0
FOP Version:	0.95
Batik Version:	1.7
Delt Report Generator Methods Version:	3.0

TBO Modules

TBO module for dlt_report_definition:	nl.delt.ReportGenerator.TBO.DltReportDefinition (Version: 3.0)
TBO module for dlt_report_category:	nl.delt.ReportGenerator.TBO.DltReportCategory (Version: 3.0)
TBO module for dlt_report_schedule:	nl.delt.ReportGenerator.TBO.DltReportSchedule (Version: 3.0)

License Information

License ID:	2
Valid repositories:	
Repository ID	**Maximum User Count**
1-16777215	2147483647
Current repository:	
1	1
Activation date (mm/dd/yyyy):	01/01/2011
Expiration date (mm/dd/yyyy):	02/10/2011

Webapplication Information

DFC Version:	6.7.0000.0213
Java Version:	Sun Microsystems Inc. 1.6.0_24
Java Total Memory:	519110656
Java Free Memory:	386873216

Operating System:	Windows Server 2008 6.0
Architecture:	x86
Delt Report Generator Webtop Version:	3.0

6 Demo reports

The euroscript Documentum Report Generator includes several demo report definitions that can create several basic reports about the repository.

The following demo Report Definitions are available:

Report Definition	Description
ACL Summary	Displays the number of ACLs, the number of internal ACLs, the number of external system ACLs and the number of external private ACLs.
Content Size by Format	Displays the average, largest and total content size per format.
Content Size by Rendition	Displays the average, largest and total content size of renditions per format.
Content Size(KB) Summary	Displays the average, largest and total content size per file store.
Number of Documents by Format	Displays the number of documents per format.
Number of Documents by Object Type [4) 5)]	Displays the number of documents per object type.
Number of Documents by Storage Area	Displays the number of documents per storage area.
Number of Groups and Users	Displays the number of groups and users in the Repository.
Jobs	Displays the name, the type, the run interval, the run mode, whether or not the job is active, the last run date and the last message of the jobs in the Repository.
Registered Tables	Displays the table name, the table owner and the owner, group and world permits of the registered tables in the Repository.
Repository configuration	Displays a selection of attributes of the dm_docbase_config object.
Server Configuration	Displays a selection of attributes of the dm_server_config object.
Server Locations	Displays physical file system paths of the dm_location objects.
Halted Workflows	Displays the workflow template name, the instance name, the supervisor, the user the currently active task was assigned to and the timestamp of the currently active task for the halted workflows in the Repository.
Number of Workflows by Supervisor	Displays the number of workflows per supervisor.

Running Workflows	Displays the number of running instances per workflow template.
Workload by Group	Displays the workload (number of inbox items) of all of the groups in the Repository.
Documents Created by me [2)]	Displays the document name, the document creator, the modify date, the current state of the documents created by the currently logged in user. The report contains direct links to the objects in the repository.
Documents Owned by me [2)]	Displays the document name, the document creator, the modify date and the current state of the documents owned by the currently logged in user. The report contains direct links to the objects in the repository.
Documents Created by a user [2) 4)]	Displays the document name and the creation date of the documents created by the specified user and from the specified date. The user running the report can select the user and the from date. (This report definition uses variables and cannot be run from a schedule because no default values are set). The report contains direct links to the objects in the repository.
My Workflows	Displays the workflow template name, the instance name, the user the currently active task was assigned to, the timestamp of the currently active task and the workflow state for the workflows of the current user.
Document Transformation Service Instances [1)]	Displays the instance name, the version and the host name of the cts_instance_info objects in the Repository.
Document Transformation Service Queue Summary [1)]	Displays the number of waiting documents and the oldest waiting item in the DTS queue.
Full Text Index Agents	Displays the index agent name, the index name, whether or not the agent was forced inactive, and whether or not the agent is active for the index agents in the Repository.
Full Text Index Queue Summary	Displays the number of waiting documents, the number of failed documents, the number of warnings and the oldest waiting item in the full text index queue.
Workload by User	Displays the number of inbox items per user.
Tasks by Work Queue	Displays the work queue category, the work queue name, the number of assigned tasks and the number of unassigned tasks for all work queues in the Repository.
Workload by Priority	Displays the workload per priority level.
Workload by Work Queue by Priority	Displays the work queue category, the work queue name, the workload and the priority for all work queues in the Repository.
Full Report [3)]	Combines most of the other reports in one report.

Tasks Assigned to me 1)	Displays the workflow name, the task name, the priority, the work queue name, the creation date and the due date for tasks assigned to the currently logged in user.
Tasks Assigned a Group or Work Queue	Displays the workflow name, the task name, the priority, to whom the task is assigned, the work queue name, the creation date and the due date for tasks assigned to a group or work queue that the currently logged in user is a member of.

1) These demo report definitions will only work if Document Transformation Services is installed. These report definitions give an error when DTS is not installed.

2) These demo reports generate links to Webtop but this only works if Webtop run on the address http://localhost:8080/webtop. To change this address you can change the style sheet for this report definition by changing the value for '<xsl:variable name="webtop">'.

3) These demo reports are available in HTML, Excel and PDF format

4) These demo reports are available in HTML and PDF format

5) The PDF format for this report can also display historic data of this report if enabled. Check "Save Report" option for the DQL Select query on this report definition to enable historic data.

Some general stylesheets are also included:

Style sheet	Description
/<cabinet>/Presentations/General Presentation.xsl	This stylesheet can be used to create a HTML file which displays the results of each query in a table.
/<cabinet>/Presentations/General Presentation CSV.xsl	This stylesheet can be used to create a CSV file with the results from the queries. It uses a comma to separate values but another separator can be used by adding an eDRG variable $separator$ to the Report Definition and setting its value to the correct separator character.
/<cabinet>/Presentations/General Presentation Excel.xsl	This stylesheet can be used to create an Excel sheet which displays the results of each query in a table form within the Excel sheet.
/<cabinet>/Presentations/General Presentation PDF.xsl	This stylesheet can be used to create a PDF file which displays the results of each query in a table.
/<cabinet>/Presentations/General Presentation Word.xsl	This stylesheet can be used to create a MS Word file which displays the results of each query in a table.
/<cabinet>/Presentations/Demo Reports/Include/charts.xsl	This stylesheet can be used to create charts in a PDF file. See paragraph 4.8.3.
/<cabinet>/Presentations/Demo Reports/Include/htmlCommon.xsl	This stylesheet is used by all the style sheets that create HTML files to create the common parts.

NOTE

These general presentations files are only installed if the demo reports are installed.

7 System Administration

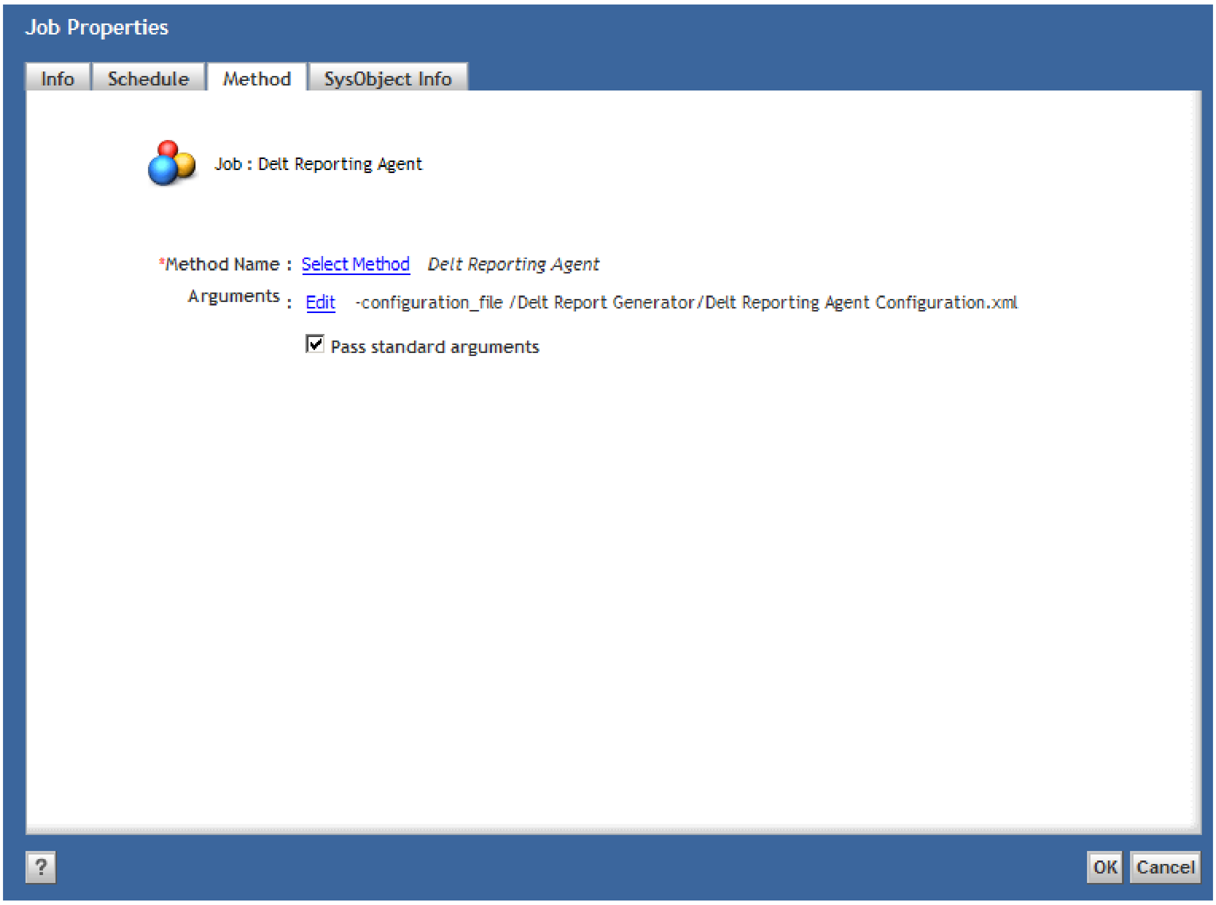
7.1 WebTop and TaskSpace Configuration

The WebTop and Taskspace integrations have one configuration file: dlt_report_generator_configuration.xml.
This file can be found in the folder \drg\config\report_generator under WebTop or TaskSpace.

It contains the following settings:

Setting	Description	Default value
ConfigurationPath	The full path in the Repository for the location where all files for the euroscript Documentum Report Generator are stored.	/Delt Reporting Configuration
method_name	Name of the method that starts the Delt Reporting Agent (should never be changed).	Delt Reporting Agent
method_config_file	The configuration file that should be used by the Delt Reporting Agent (must be the same as the value for –configuration_file on the job settings).	/Delt Reporting Configuration/Delt Reporting Agent Configuration.xml
method_trace_level	The trace level that the Delt Reporting Agent uses when the users start it manually. If set to 0 (zero), the log file will not be created but you will not get any error messages if something goes wrong either.	2

7.2 Job Configuration



To change the job settings for the Delt Report Generator:

- Login to Documentum Administrator.
- Navigate to Administration\Job Management\Jobs.
- Locate the Delt Reporting Agent job.
- Click the link Select Method on the Method tab.
- Specify the arguments.

Pass standard arguments must always be checked. There is one additional custom argument 'configuration_file' for this job. This parameter must be specified on a single line. It should point to the full location in the repository of the configuration file for the job. The default location for the configuration file '**Delt Reporting Agent Configuration.xml**' is '**/Delt Reporting Configuration**'.
Example: "-configuration_file /Delt Reporting Configuration/Delt Reporting Agent Configuration.xml".

The configuration file contains the following settings:

Setting	Description	Default value
license_file	The full location to the license file on the file system.	/{eDRG Cabinet}/eDRG.license
language	The language used for messages in the log files. The language setting may only be changed if the translated messages are available in this file.	En
max_instances	The maximum number of allowed reports to be generated concurrently.	10
allow_historic_data	Indicates if historic data is allowed for this repository.	True
allow_auto_create_historic_tables	Indicates if historic data tables can be automatically created/updated when needed.	True
save_usage_info	Indicates if the eDRG usage information is saved in the repository.	True

7.3 Changing the eDRG Cabinet

By default, the euroscript Documentum Report Generator stores its files in the cabinet Delt Report Configuration.
It is possible to change the name of this cabinet.

To change the name of the Delt Report Configuration cabinet:

- Login to Documentum Webtop.
- Navigate to **Cabinets**.
- View the properties of the cabinet Delt Report Configuration.
- Change the name.
- Save the cabinet object.

On the application server:

- Navigate to the Webtop or TaskSpace folder structure.
- Navigate to \drg\config\report_generator\dlt_report_generator.
- Open the file dlt_report_generator_configuration.xml.
- Change the value between the <ConfigurationPath> tags to reflect the new cabinet name.
- Change the value between the <method_config_file> tags to reflect the new cabinet name.
- Save the file.
- Restart the application server.

or
Go to this page: http://<application-server>/webtop/wdk/refresh.jsp.

7.4 Checking Method Logs

Each time when a report is generated, a log file is created (unless the trace level is set to 0). If the report is generated manually or the schedule is started with the [**Test now**] button, the log file is placed in the Temp cabinet. If the Delt Reporting Agent job is run, the log file is created in **/Temp/jobs/Delt Reporting Agent**.

Messages that can appear in the log file:

Message	Description
Schedule: Report definition: DQL Query: Presentation file: Destination:	This is normally shown after an error occurred. It gives you more information on where the error occurred.
ERROR: No license file present	The license specified in the configuration file cannot be found. *)
ERROR: Wrong license file version	The license file is for a different version of the euroscript Documentum Report Generator. *)
ERROR: Corrupt license file	The license file is either damaged or the file specified in the configuration file is not a license file. *)
ERROR: Current repository does not fall within the license Too many users in repository	Your repository has more report users then the current license file allows. *)
ERROR: Current repository does not fall within the license Docbase ID not in licensed range	The docbase id of your repository is not licensed to use the euroscript Documentum Report Generator.*)
ERROR: Current repository does not fall within the license Licensed period exceeded	The license for the euroscript Documentum Report Generator has either expired or has not begun yet. *)
ERROR: Error while reading license file	There was an error reading the license file. After this message the error will be displayed. *)
ERROR: No license file specified.	The location of the license file is not specified in the configuration file.
ERROR: Required argument user_name missing. Make sure that 'pass standard arguments' is checked on the job settings.	Something is wrong with the job settings. Check these settings.
ERROR: Required argument docbase_name missing. Make sure that 'pass standard arguments' is checked on the job settings.	
ERROR: The arguments report_definitions and test_schedule are not valid when this method is started by a job	
ERROR: The arguments report_definitions and test_schedule cannot be specified together	
Current DQL query is not a select query. Other query types are not allowed	The specified DQL query is not a SELECT query, check the query that is listed below this message.
Unknown destination type only Repository, OS and	These errors are only possible if the

<table>
<tr><td>Email are allowed</td><td rowspan="2">values on the report definition are incorrect. Open the report definition, edit all destinations or presentations and save the object.</td></tr>
<tr><td>Unknown presentation type only Repository and OS are allowed</td></tr>
<tr><td>Cannot find a matching 0-argument function named {http://xml.apache.org/xslt/java}java.util.Date.new();</td><td>Messages like this mean that the stylesheet uses Xalan specific functions while the Saxon XSLT processor is installed.</td></tr>
<tr><td>The historic info table 'edrg_history_info' does not exist and creating it is not allowed.</td><td rowspan="2">Some historic data tables don't exist or are incorrect and automatic creation is disabled.</td></tr>
<tr><td>Historic data tables for this report definition don't exist and creating it is not allowed.</td></tr>
<tr><td>Some queries on this report have the save result option enabled but saving history data has been disabled for this repository.</td><td>This is only a warning. The report definition has save reports enabled but the historic data option is globally disabled. The report is generated but the results are not stored and no historic data is included.</td></tr>
</table>

*) A request for a new or updated license file should be directed to euroscript by submission of a request form. The license file will be sent by e-mail.

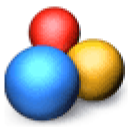
7.5 ACL's

The euroscript Documentum Report Generator uses six permission sets.

Description of the permission sets:

ACL	Description
dlt_report_cabinet_acl	This ACL is placed on the cabinet that contains most of the files related to the euroscript Documentum Report Generator.
dlt_report_configuration_acl	This ACL is placed on the configuration xml file in the Delt Reporting Configuration cabinet.
dlt_report_definition_acl	This ACL is placed on the new report categories and report definitions by default (unless the parent category has a different ACL, in which case the different ACL is used).
dlt_report_schedule_acl	This ACL is placed on all new report schedules.
dlt_report_temp_folder_acl	This ACL is placed on the folder that contains temporary files created by the euroscript Documentum Report Generator
dlt_report_agent_acl	This ACL is placed on the eDRG method.

7.6 Job/Method



The euroscript Documentum Report Generator uses a Job and a Method to generate the reports.

- Job: Delt Reporting Agent
- Method: Delt Reporting Agent

The Job is used for periodically processing scheduled reports. The Method is called by the Job as well as when a user manually requests a report.

Another Job: "Delt Cleanup History" is used to periodically remove old reports from the Historic Data Tab.

Appendix I. Java formats

Valid Java date formats (taken from Java documentation).

The following pattern letters can be used:

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800

Pattern letters are usually repeated, as their number determines the exact presentation:

Text: for formatting, if the number of pattern letters is 4 or more, the full form is used. Otherwise, if available, a short or abbreviated form is used. For parsing, both forms are accepted, independent of the number of pattern letters.

Number: for formatting, if the number of pattern letters is the minimum number of digits, and shorter numbers are zero-padded to this amount. For parsing, the number of pattern letters is ignored unless it is needed to separate two adjacent fields.

Year: for formatting, if the number of pattern letters is 2 then the year is truncated to 2 digits. Otherwise, it is interpreted as a number.

For parsing, if the number of pattern letters is more than 2, the year is interpreted literally, regardless of the number of digits. So using the pattern "MM/dd/yyyy", "01/11/12" parses to Jan 11, 12 A.D.

Examples
The following examples show how date and time patterns are interpreted in the U.S. locale. The given date and time are 2001-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Date and Time Pattern	Result
"yyyy.MM.dd G 'at' HH:mm:ss z"	2001.07.04 AD at 12:08:56 PDT
"EEE, MMM d, ''yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o''clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02001.July.04 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 4 Jul 2001 12:08:56 -0700
"yyMMddHHmmssZ"	010704120856-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSZ"	2001-07-04T12:08:56.235-0700

Valid java number formats (from Java documentation):

Symbol	Location	Meaning
0	Number	Digit.
#	Number	Digit, zero shows as absent .
.	Number	Decimal separator or monetary decimal separator .
-	Number	Minus sign.
,	Number	Grouping separator.
E	Number	Separates mantissa and exponent in scientific notation. Need not be quoted in prefix or suffix.
;	Sub pattern boundary	Separates positive and negative sub patterns.
%	Prefix or suffix	Multiply by 100 and show as percentage.
\u2030	Prefix or suffix	Multiply by 1000 and show as per mille.
¤ (\u00A4)	Prefix or suffix	Currency sign, replaced by currency symbol. If doubled, replaced by international currency symbol. If present in a pattern, the monetary decimal separator is used instead of the decimal separator.
'	Prefix or suffix	Used to quote special characters in a prefix or suffix, for example, "'#'#" formats 123 to "#123". To create a single quote itself, use two in a row: "# o''clock".